2nd Conference on Language, Data and Knowledge

LDK 2019, May 20–23, 2019, Leipzig, Germany

Edited by

Maria Eskevich
Gerard de Melo
Christian Fäth
John P. McCrae
Paul Buitelaar
Christian Chiarcos
Bettina Klimek
Milan Dojchinovski

Editors

Maria Eskevich
CLARIN ERIC, Utrecht, The Netherlands
maria@clarin.eu

Gerard de Melo
Department of Computer Science, Rutgers University–New Brunswick, NJ, USA

Christian Fäth
Applied Computational Linguistics, Goethe University Frankfurt, Germany

John P. McCrae
Insight Centre for Data Analytics, National University of Ireland Galway, Ireland

Paul Buitelaar
Insight Centre for Data Analytics, National University of Ireland Galway, Ireland

Christian Chiarcos
Applied Computational Linguistics, Goethe University Frankfurt, Germany

Bettina Klimek
Agile Knowledge Engineering and Semantic Web, University of Leipzig, Germany

Milan Dojchinovski
Agile Knowledge Engineering and Semantic Web, University of Leipzig, Germany

ACM Classification 2012
Computing methodologies → Natural language processing; Computing methodologies → Knowledge representation and reasoning

ISBN 978-3-95977-105-4

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at https://www.dagstuhl.de/dagpub/978-3-95977-105-4.

Publication date
May, 2019

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at https://portal.dnb.de.

License
This work is licensed under a Creative Commons Attribution 3.0 Unported license (CC-BY 3.0):
https://creativecommons.org/licenses/by/3.0/legalcode.
In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:
- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/OASIcs.LDK.2019.0
ISBN 978-3-95977-105-4 **ISSN 1868-8969** **https://www.dagstuhl.de/oasics**

OASIcs – OpenAccess Series in Informatics

OASIcs aims at a suitable publication venue to publish peer-reviewed collections of papers emerging from a scientific event. OASIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Editorial Board

- Daniel Cremers (TU München, Germany)
- Barbara Hammer (Universität Bielefeld, Germany)
- Marc Langheinrich (Università della Svizzera Italiana – Lugano, Switzerland)
- Dorothea Wagner (*Editor-in-Chief*, Karlsruher Institut für Technologie, Germany)

ISSN 1868-8969

https://www.dagstuhl.de/oasics

Contents

Preface
Maria Eskevich, Gerard de Melo, Christian Fäth and Christian Chiarcos 0:ix

Foundations: Web Technologies and Vocabularies

SPARQL Query Recommendation by Example: Assessing the Impact of Structural Analysis on Star-Shaped Queries
Alessandro Adamou, Carlo Allocca, Mathieu d'Aquin, and Enrico Motta 1:1–1:8

OWL^C: A Contextual Two-Dimensional Web Ontology Language
Sahar Aljalbout, Didier Buchs, and Gilles Falquet 2:1–2:13

Ligt: An LLOD-Native Vocabulary for Representing Interlinear Glossed Text as RDF
Christian Chiarcos and Maxim Ionov 3:1–3:15

The Shortcomings of Language Tags for Linked Data When Modeling Lesser-Known Languages
Frances Gillis-Webber and Sabine Tittel 4:1–4:15

Functional Representation of Technical Artefacts in Ontology-Terminology Models
Laura Giacomini 5:1–5:6

Language and Data: Human Language Technology

Comparison of Different Orthographies for Machine Translation of Under-Resourced Dravidian Languages
Bharathi Raja Chakravarthi, Mihael Arcan, and John P. McCrae 6:1–6:14

CoNLL-Merge: Efficient Harmonization of Concurrent Tokenization and Textual Variation
Christian Chiarcos and Niko Schenk 7:1–7:14

Exploiting Background Knowledge for Argumentative Relation Classification
Jonathan Kobbe, Juri Opitz, Maria Becker, Ioana Hulpuş, Heiner Stuckenschmidt, and Anette Frank 8:1–8:14

Graph-Based Annotation Engineering: Towards a Gold Corpus for Role and Reference Grammar
Christian Chiarcos and Christian Fäth 9:1–9:11

Crowd-Sourcing A High-Quality Dataset for Metaphor Identification in Tweets
Omnia Zayed, John P. McCrae, and Paul Buitelaar 10:1–10:17

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Data and Knowledge: Entities and Relations

Inflection-Tolerant Ontology-Based Named Entity Recognition for Real-Time Applications
Christian Jilek, Markus Schröder, Rudolf Novik, Sven Schwarz, Heiko Maus, and Andreas Dengel 11:1–11:14

Validation Methodology for Expert-Annotated Datasets: Event Annotation Case Study
Oana Inel and Lora Aroyo 12:1–12:15

A Proposal for a Two-Way Journey on Validating Locations in Unstructured and Structured Data
Ilkcan Keles, Omar Qawasmeh, Tabea Tietz, Ludovica Marinucci, Roberto Reda, and Marieke van Erp 13:1–13:8

Name Variants for Improving Entity Discovery and Linking
Albert Weichselbraun, Philipp Kuntschik, and Adrian M. P. Braşoveanu 14:1–14:15

Interlinking SciGraph and DBpedia Datasets Using Link Discovery and Named Entity Recognition Techniques
Beyza Yaman, Michele Pasin, and Markus Freudenberg 15:1–15:8

Language and Knowledge: Lexical Data

lemon-tree: Representing Topical Thesauri on the Semantic Web
Sander Stolk 16:1–16:13

Translation-Based Dictionary Alignment for Under-Resourced Bantu Languages
Thomas Eckart, Sonja Bosch, Dirk Goldhahn, Uwe Quasthoff, and Bettina Klimek 17:1–17:11

Cherokee Syllabary Texts: Digital Documentation and Linguistic Description
Jeffrey Bourns 18:1–18:6

Metalexicography as Knowledge Graph
David Lindemann, Christiane Klaes, and Philipp Zumstein 19:1–19:8

Cross-Dictionary Linking at Sense Level with a Double-Layer Classifier
Roser Saurí, Louis Mahon, Irene Russo, and Mironas Bitinis 20:1–20:16

Towards the Detection and Formal Representation of Semantic Shifts in Inflectional Morphology
Dagmar Gromann and Thierry Declerck 21:1–21:15

Applications in the Language Sciences

Opening Digitized Newspapers Corpora: Europeana's Full-Text Data Interoperability Case
Nuno Freire, Antoine Isaac, Twan Goosen, Daan Broeder, Hugo Manguinhas, and Valentine Charles 22:1–22:14

Automatic Detection of Language and Annotation Model Information in CoNLL Corpora
Frank Abromeit and Christian Chiarcos 23:1–23:9

The Secret to Popular Chinese Web Novels: A Corpus-Driven Study
Yi-Ju Lin and Shu-Kai Hsieh 24:1–24:8

Predicting Math Success in an Online Tutoring System Using Language Data and Click-Stream Variables: A Longitudinal Analysis
Scott Crossley, Shamya Karumbaiah, Jaclyn Ocumpaugh, Matthew J. Labrum, and Ryan S. Baker 25:1–25:13

Can Computational Meta-Documentary Linguistics Provide for Accountability and Offer an Alternative to "Reproducibility" in Linguistics?
Tobias Weber 26:1–26:8

Preface

This volume presents the proceedings of the 2nd Conference on Language, Data and Knowledge (LDK 2019) held in Leipzig, Germany, May 20–23, 2019. Language, Data and Knowledge is a bi-annual conference series on matters of human language technology, data science, and knowledge representation, initiated in 2017 by a consortium of researchers from the Insight Centre for Data Analytics at the National University of Ireland, Galway (Ireland), the Institut für Angewandte Informatik (InfAI) at the University of Leipzig (Germany), and the Applied Computational Linguistics Lab (ACoLi) at Goethe University Frankfurt am Main (Germany), and it has been supported by an international Scientific Committee of leading researchers in Natural Language Processing, Linked Data and Semantic Web, Language Resources and Digital Humanities.

The second edition of the LDK conference is hosted by the Institut für Angewandte Informatik (InfAI) in Leipzig, Germany and co-organized by the Insight Centre for Data Analytics and the Applied Computational Linguistics Lab (ACoLi). Major Sponsors were the *LiLa: Linking Latin* project, the *CID GmbH* in Germany, the *Semantic Web Company*, and *Pret-a-LLOD. Ready-to-use Multilingual Linked Language Data for Knowledge Services across Sectors* funded under the European Union's Horizon research and innovation programme under grant agreement No. 825182. LDK 2019 has received further endorsement from the *DBpedia Association*, from the *European Lexicographic Infrastructure (ELEXIS)* project funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 731015, and from the independent research group *Linked Open Dictionaries (LiODi)* funded by the German Federal Ministry of Education and Research (BMBF).

In a biennial cycle, LDK conferences aim at bringing together researchers from across disciplines concerned with the acquisition, curation and use of language data in the context of data science and knowledge-based applications. With the advent of the Web and digital technologies, an ever increasing amount of language data is now available across application areas and industry sectors, including social media, digital archives, company records, etc. The efficient and meaningful exploitation of this data in scientific and commercial innovation is at the core of data science research, employing natural language processing and machine learning methods as well as semantic technologies and knowledge graphs.

Language data is of increasing importance to machine learning-based approaches in Human Language Technologies, Linked Data and Semantic Web research and applications that depend on linguistic and semantic annotation with lexical, terminological and ontological resources, manual alignment across language or other human-assigned labels. The acquisition, provenance, representation, maintenance, usability, quality as well as legal, organizational and infrastructure aspects of language data are therefore rapidly becoming major areas of research that are at the focus of the conference.

Knowledge graphs is an active field of research concerned with the extraction, integration, maintenance and use of semantic representations of language data in combination with semantically or otherwise structured data, numerical data and multimodal data among others. Knowledge graph research builds on the exploitation and extension of lexical, terminological and ontological resources, information and knowledge extraction, entity linking, ontology learning, ontology alignment, semantic text similarity, Linked Data and other Semantic Web technologies. The construction and use of knowledge graphs from language data, possibly and ideally in the context of other types of data, is a further specific focus of the conference.

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

As in previous years, LDK 2019 features a number of collocated satellite events dedicated to the conference topics. This includes the 13th DBpedia community meeting, the 2nd Shared Task on Translation Inference Across Dictionaries (TIAD), a workshop of the W3C Ontology-Lexica Community and Business Group and a tutorial on historical text reuse (TRACER).

In addition, this edition of LDK also features an associated summer school, the 3rd Summer Datathon on Linguistic Linked Open Data (SD-LLOD-19, held in Schloss Dagstuhl – Leibniz Center for Informatics, Wadern, Germany), which complements the scientific focus of the conference with a didactic component and a hands-on experience. The SD-LLOD datathon has the main goal of giving people from industry and academia practical knowledge in the field of Linked Data and its application to natural language data and natural language annotations, from areas as diverse as knowledge engineering, lexicography, the language sciences, natural language processing and computational philology.

In total, 43 papers were submitted and reviewed by 88 reviewers. Typically, at least 3 reviews per paper resulted in 26 accepted papers. As a novel feature, LDK-2019 had a special track for short abstracts on latest development to be presented as posters during the conference. However, these are not subject to the proceedings and will be published separately.

The conference programme additionally encompasses invited talks on *Mapping the Lexicons of Signs and Words* by Christiane Fellbaum (Princeton University), and on *Schema.org Annotations and Web Tables: Underexploited Semantic Nuggets on the Web?* by Christian Bizer (Mannheim University), as well as on *The Sorbian languages* by Eduard Werner (University of Leipzig).

Organizing Committee

Conference Chairs

John P. McCrae (National University of Ireland Galway)
Paul Buitelaar (National University of Ireland Galway)
Christian Chiarcos (Goethe University Frankfurt)

Local Organizers

Bettina Klimek (University of Leipzig)
Milan Dojchinovski (University of Leipzig)

Program Chairs

Gerard de Melo (Rutgers University)
Maria Eskevich (CLARIN ERIC)

Proceedings Chair

Christian Fäth (Goethe University Frankfurt)

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Scientific Advisory Committee

Francis Bond (Nanyang Technological University)
Paul Buitelaar (National University of Ireland Galway)
Christian Chiarcos (Goethe University Frankfurt)
Philipp Cimiano (Bielefeld University)
Edward Curry (National University of Ireland Galway)
Thierry Declerck (Deutsches Forschungszentrum für Künstliche Intelligenz – DFKI)
Milan Dojchinovski (University of Leipzig)
Tatjana Gornostaja (Tilde)
Jorge Gracia (University of Zaragoza)
Nancy Ide (Vassar College)
Franciska de Jong (CLARIN ERIC)
John P. McCrae (National University of Ireland Galway)
Gerard de Melo (Rutgers University)
Karin Verspoor (University of Melbourne)

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Program Committee

Eneko Agirre (University of the Basque Country)

Malihe Alikhani (Rutgers University)

Sören Auer (TIB Leibniz Information Center Science & Technology and University of Hannover)

Denilson Barbosa (University of Alberta)

Pierpaolo Basile (Dipartimento di Informatica – University of Bari)

Valerio Basile (University of Turin)

Martin Benjamin (Kamusi Project International)

Joanna Biega (Max Planck Institute for Informatics)

Michael Bloodgood (The College of New Jersey)

Francis Bond (Nanyang Technological University)

Harry Bunt (Tilburg University)

Aljoscha Burchardt (Deutsches Forschungszentrum für Künstliche Intelligenz – DFKI)

Nicoletta Calzolari (Istituto di Linguistica Computazionale – CNR)

Richard Eckart de Castilho (Ubiquitous Knowledge Processing Lab – UKP)

Philipp Cimiano (Bielefeld University)

Francesco Corcoglioniti (Fondazione Bruno Kessler)

Patrick Ernst (Max-Planck Institute for Informatics)

Besnik Fetahu (L3S Research Center)

Darja Fišer (University of Ljubljana)

Francesca Frontini (Université Paul-Valéry Montpellier 3 Praxiling UMR 5267 CNRS – UPVM3)

Luis Galárraga (Aalborg University)

Debanjan Ghosh (Rutgers University)

Jeff Good (University at Buffalo)

Gregory Grefenstette (IHMC and Biggerpan Inc)

Dagmar Gromann (TU Dresden)

Graeme Hirst (University of Toronto)

Eero Hyvönen (Aalto University and University of Helsinki – HELDIG)

Nancy Ide (Vassar College)

Sepehr Janghorbani (Rutgers University)

Richard Johansson (University of Gothenburg)

Te Taka Keegan (University of Waikato)

Roman Klinger (University of Stuttgart)

Dimitris Kontokostas (University of Leipzig)

Maria Koutraki (Leibniz Institute for Information Infrastructure – FIZ, and Karlsruher Institut für Technologie – KIT)

Udo Kruschwitz (University of Essex)

Oi Yee Kwong (The Chinese University of Hong Kong)

Piroska Lendvai (University of Göttingen)

Margot Mieskes (University of Applied Sciences Darmstadt)

Pasquale Minervini (University of Bari)

Paramita Mirza (Max Planck Institute for Informatics)

Elena Montiel-Ponsoda (Universidad Politécnica de Madrid)

Steven Moran (University of Zurich)

Andrea Moro (Sapienza University of Rome)

Hugo Gonçalo Oliveira (University of Coimbra)

Alessandro Oltramari (Bosch Research and Technology Center)

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Petya Osenova (Sofia University, and Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences – IICT-BAS)

Sebastian Pado (University of Stuttgart)

Natalie Parde (University of Illinois at Chicago)

Bolette Pedersen (University of Copenhagen)

Pascual Pérez-Paredes (University of Cambridge)

Maciej Piasecki (Department of Computational Intelligence Wroclaw University of Science and Technology)

Laurette Pretorius (School of Interdisciplinary Research and Graduate Studies University of South Africa)

Gábor Prószéky (Pázmány Péter Catholic University Budapest)

Francesca Quattri (The Hong Kong Polytechnic University)

Alexandre Rademaker (IBM Research Brazil and EMAp/FGV)

Dheeraj Rajagopal (Carnegie Mellon University)

Simon Razniewski (Max Planck Institute for Informatics)

Georg Rehm (Deutsches Forschungszentrum für Künstliche Intelligenz – DFKI)

Nils Reiter (Institute of Natural Language Processing Stuttgart University)

Steffen Remus (University of Hamburg)

Laurent Romary (Institut national de recherche en informatique et en automatique – INRIA, and HUB-ISDL)

Marco Rospocher (Università degli Studi di Verona)

Harald Sack (Leibniz Institute for Information Infrastructure – FIZ, and Karlsruher Institut für Technologie – KIT)

Felix Sasaki (Lambdawerk)

Andrea Schalley (Karlstad University)

Gilles Serasset (LIG Université Grenoble I)

Vered Shwartz (Bar-Ilan University)

Max Silberztein (Université de Franche-Comté)

Aitor Soroa (Universidad del País Vasco / Euskal Herriko Unibertsitatea – UPV/EHU)

Steffen Staab (Institut WeST University Koblenz-Landau and WAIS University of Southampton)

Armando Stellato (University of Rome Tor Vergata)

Stan Szpakowicz (University of Ottawa)

Niket Tandon (Max Planck Institute for Informatics)

Sara Tonelli (Fondazione Bruno Kessler)

Mihaela Vela (Universität des Saarlandes)

Marc Verhagen (Brandeis University)

Karin Verspoor (The University of Melbourne)

Piek Vossen (Vrije Universiteit Amsterdam)

Sabine Schulte Im Walde (University of Stuttgart)

Ulli Waltinger (Siemens AG – Corporate Technology)

Eveline Wandl-Vogt (Austrian Centre for Digital Humanities @ Austrian Academy of Sciences – ACDH)

Linlin Wang

Aaron Steven White (University of Rochester)

Michael Witbrock (IBM)

René Witte (Concordia University)

Qian Yang (Tsinghua University)

Kalliopi Zervanou (Eindhoven University of Technology)

Ziqi Zhang (Sheffield University)

SPARQL Query Recommendation by Example: Assessing the Impact of Structural Analysis on Star-Shaped Queries

Alessandro Adamou
Data Science Institute, National University of Ireland Galway, Ireland
alessandro.adamou@nuigalway.ie

Carlo Allocca
Samsung Inc., London, United Kingdom
c.allocca@samsung.com

Mathieu d'Aquin
Data Science Institute, National University of Ireland Galway, Ireland
mathieu.daquin@nuigalway.ie

Enrico Motta
Knowledge Media Institute, The Open University, Milton Keynes, United Kingdom
enrico.motta@open.ac.uk

Abstract

One of the existing query recommendation strategies for unknown datasets is "by example", i.e. based on a query that the user already knows how to formulate on another dataset within a similar domain. In this paper we measure what contribution a structural analysis of the query and the datasets can bring to a recommendation strategy, to go alongside approaches that provide a semantic analysis. Here we concentrate on the case of star-shaped SPARQL queries over RDF datasets.

The illustrated strategy performs a least general generalization on the given query, computes the specializations of it that are satisfiable by the target dataset, and organizes them into a graph. It then visits the graph to recommend first the reformulated queries that reflect the original query as closely as possible. This approach does not rely upon a semantic mapping between the two datasets. An implementation as part of the SQUIRE query recommendation library is discussed.

2012 ACM Subject Classification Information systems → Semantic web description languages

Keywords and phrases SPARQL, query recommendation, query structure, dataset profiling

Digital Object Identifier 10.4230/OASIcs.LDK.2019.1

Category Short Paper

Acknowledgements This work was supported by the MK:Smart project (OU Ref. HGCK B4466).

1 Introduction

One of the main characteristics of the Linked Open Data (LOD) cloud is the heterogeneity of schemas and vocabularies: it is not rare for RDF datasets to use different vocabularies to describe similar domains. However, this also entails a proliferation of ontologies for overlapping domains appearing across datasets. For example, universities and academic institutions intermittently use AIISO[1] or XCRI[2] to describe their courses in RDF, and occasionally use the same properties in different ways. This increases the difficulty to build, for instance, a recommender system for courses offered by all the universities in

[1] Academic Institution Internal Structure Ontology, `http://purl.org/vocab/aiiso/schema#`
[2] eXchanging Course-Related Information, `http://xcri.org/`


© Alessandro Adamou, Carlo Allocca, Mathieu d'Aquin, and Enrico Motta;
licensed under Creative Commons License CC-BY

a country. Finding the right queries for all the datasets usually involves intensive and time-consuming preliminary work to explore and understand each dataset's data model and content, then iteratively reformulate and test SPARQL queries [13]. However, if the user has prior knowledge of some of the datasets and a tool that can exploit such knowledge to aid them in the reformulation, this effort can be reduced. Semantic analysis techniques such as ontology alignment provide support to querying unknown datasets [11], however most of them only consider how the terms are defined in the corresponding ontologies, hardly considering their roles in the datasets in terms of relationships between the structure of the query and that of the dataset. We investigate how much query recommendation "by example" can benefit from structural analysis for recommending the best queries soonest.

We propose a method that, given a SPARQL query q_o that a source RDF dataset D^S is able to answer, reformulates it into queries that can be answered by a target RDF dataset D^T and reflect as closely as possible the intended meaning, structure and types of results of the original query. The approach analyzes the structure of the query and relates it to the schema of both datasets, by computing a *least general generalization* and navigating the resulting operational structure of its specializations. It does not require an ontology mapping and/or instance matching between the datasets, but such methods can be integrated with it, by influencing the recommendation ranking. Basic constructs of RDF(S) and OWL are considered, but only as potential invariants in the query: in other words, the signature of a class is merely treated like a structure. In this paper, we concentrate on *star-shaped* queries, a class of queries that still represents a significant challenge in query optimization [8, 12].

The implementation of the method has been included with an open source SPARQL query recommendation toolkit called SQUIRE [1], which comes as a software library, standalone program and Web application. This offers valuable support for query recommendation, empowering not only the automatic learning of the data model and content of an RDF dataset, but also its straightforward use without the user's prior knowledge of its content.

After illustrating related work in Section 2, the method is detailed in Section 3. Section 4 provides some insight on the experiments underway, before concluding with future work.

2 Related Work

Most research on SPARQL query recommendation uses ontology alignments or schema mappings explicitly defined between the source and target dataset [5, 10]. We are not aware of any study on how to recommend queries over unmapped datasets based not on a full schema, but on just enough knowledge of one to write example queries over another dataset.

Our work was partly inspired by the intuition that the capabilities of a dataset, in terms of what questions it is able to answer, can be understood by organizing them into a data structure that can be navigated through methods such as formal concept analysis [6].

The employment of least general generalizations is not new in semi-automated SPARQL querying: we acknowledge that Lehmann et al have previously implemented it with success for OWL-based machine learning in DL-Learner [3].

Although the above studies contribute interesting elements for us to build on, they were mainly driven either by the user's lack of familiarity with the underlying technologies (which is not our case), or by the availability of semantic alignments. Regarding the latter, the placement of our work is immediately *before* semantic analysis takes place, in an effort to understand if such techniques can be further led to converge towards ideal recommendations.

Lastly we acknowledge the existence of effective methods, such as that by Fokou et al [7], to relax failing queries, rather than outright avoid them, for obtaining satisfiable ones. We are in fact in contact with the authors to investigate a possible intertwining of both techniques.

3 Method

A *star-shaped* SPARQL query is a query such that all its graph pattern expressions (GPEs) are either triple patterns $(subject, predicate, object)$ that share the same subject, or GPEs obtained by combining them using the AND operator. For instance, the following:

```
SELECT DISTINCT ?title ?pic
WHERE {
  ?s  rdf:type bibo:Book ; dc:title ?title ; foaf:depiction ?pic .
}
```

is a star-shaped query of 3 TPs to get the titles and pictures (e.g. front covers) of books.

To generate query recommendations by example, we first (*i*) deconstruct the original query into one or more general queries satisfiable by both datasets (*Generalization*); (*ii*) transform the general queries into queries for the target dataset (*Specialization*); and (*iii*) grade and rank the generated queries to select which ones to recommend (*Evaluation*).

3.1 Generalization Step

Suppose a query q_o is satisfiable (i.e. produces a non-empty result set) w.r.t. a dataset D^S but not necessarily D^T. The generalization step produces a set of queries $\mathcal{Q}^G = \{q_i^G, i = 1..m\}$ so that every q_i^G is satisfiable by both D^S and D^T, but if a non-redundant and non-trivial triple pattern (i.e. one that is not composed solely of variables and is not a repetition of an existing triple pattern in q_i^G) were added to it, the resulting query would no longer be satisfiable by both datasets. This is called a "least general generalization" (lgg) of q_o.

When a TP cannot be preserved in an lgg, for it would make the query no longer satisfiable by both datasets, its terms are excluded from the generalization and replaced with special, unique SPARQL variables called *template variables*. These variables use the convention ct_j, opt_j and dpt_j, respectively for the j-th class, object property and data property template variable. The sets of terms of each category for D^S and D^T are obtained by inspecting each dataset and maintaining an index of the terms. The ppt_j convention is used for "plain" (RDF) properties when it cannot be determined if they are being *used as* object or datatype properties. To reduce the risk of combinatorial explosion of recommendations, named individuals and literals in a TP are not replaced with template variables. It follows by construction that the members of an lgg are themselves queries.

Algorithm 1 below illustrates this rationale. Given the initial query q_o (*2-5*): take every class or property that appears in D^S but not in D^T and substitute every occurrence of it with an occurrence of a new template variable. At this point (*6*) we have one generalized query, but it is not guaranteed to be satisfiable by D^T. Therefore, build a property co-occurrence matrix M (*7*), which is a square matrix indexed by properties in D^T that indicates the classes, if any, where they appear together. If there are still concrete classes left, (*8-12*) add to $\mathcal{Q}^G$ one query for each of them and generalize in each query every property that does not occur (i.e. does not co-occur with itself in M) for members of that class. Now (*13*) take the queries that were generated in the steps before and for each (*14-17*) find in M the largest sets of co-occurring properties appearing in that query (P_j is the set of their indices), then add to $\mathcal{Q}^G$ one query for every such group, so that only properties of that group appear in it. If steps *7-17* did not produce any queries, then the query produced before was already general enough, therefore (*18*) make that the lgg.

Algorithm 1 Least general generalization of an input query.
INPUT: two datasets D^S and D^T; a query q_o that is satisfiable w.r.t. D^S
OUTPUT: a set $\mathcal{Q}^G$ of queries that generalize q_o and are satisfiable w.r.t. D^S and D^T

procedure GENERALIZE(q_o, D^S, D^T)
  for each non-variable node p_i of q_o **do**
    if p_i is an object property in D^S but not in D^T) **then**
      Replace every occurrence of p_i with a new object property template variable.
    Do the same for data properties, other properties and classes of D^S.
  $q_o^G \leftarrow$ the new query resulting from the above
  $M \leftarrow$ co-occurrence matrix of all the properties in D^T w.r.t classes in D^T
  for each concrete class c_i in q_o^G **do**
    Generate a new query $q_{c_i}^G$ with all the TPs of q_o^G where c_i occurs
    for each predicate p_j of q_o^G that is not `rdf:type` **do**
      if $c_i \in M_{jj}$ **then** add to $q_{c_i}^G$ all the TPs with p_j
      else add to $q_{c_i}^G$ all the corresponding TPs after generalizing p_j
  $\mathcal{Q}^G \leftarrow \{q_{c_i}^G\}_i$
  for each $q_i^G \in \mathcal{Q}^G$ if $\mathcal{Q}^G \neq \emptyset$ otherwise q_o^G **do**
    for each p_j occurring in q_i^G **do**
      $P_j \leftarrow$ largest group of properties p_k so that $M_{jk} \neq \emptyset$
      Add to $\mathcal{Q}^G$ a new query with only the TPs of q_i^G whose predicates are in P_j
  if $\mathcal{Q}^G = \emptyset$ **then** $\mathcal{Q}^G \leftarrow \{q_o^G\}$

For example, suppose the first round of generalization has produced a query:

```
SELECT DISTINCT ?title ?pic WHERE {
  ?s  rdf:type ?ct1 ; dc:title ?title ; foaf:depiction  ?pic .
}
```

and `dc:title` and `foaf:depiction` are both present in D^T but never together. In that case, the following rounds will generate two queries (the largest property groups being {`rdf:type`, `dc:title`} and {`rdf:type`, `foaf:depiction`}), whose patterns are respectively:

```
  ?s  rdf:type ?ct1 ; ?dpt1 ?title ; foaf:depiction  ?pic .

  ?s  rdf:type ?ct1 ; dc:title ?title ; ?opt1 ?pic .
```

Having computed the set that represents the lgg of the original query, we need to know in which ways it can be transformed into a set of queries over the target dataset D^T, and detect those queries that are potentially closer to the original one. This is what specialization does.

3.2 Specialization Step

The goal of specializing an lgg $\mathcal{Q}^G$ is to generate the space of candidate queries, regardless of which query is to be preferred over which. A necessary condition for a query to be among the candidates is that it must be satisfiable by the target dataset D^T.

Specialization can be regarded as the repeated application of some operation over a query. We distinguish two such operations: *Removal* (*Rop*) and *Instantiation* (*Iop*).

Removal (Rop) systematically removes an entire TP from a query, as well as removing from its projection (e.g. the SELECT statement) any variable that appeared only in that TP. Along with the obvious precondition of there being more than one TP in the query for *Rop* to be applied, we also restrict to only applying *Rop* on TPs that contain at least one template variable.

Instantiation (Iop) replaces every occurrence of a template variable in a query with one concrete (non-variable) value, thereby instantiating the template variable in question.

Applied to our specialization method, removals may be performed on a query regardless of the target dataset, since by monotonicity no solutions are lost if a TP is removed from an intersection of TPs. Instantiations, on the other hand, are performed so as to preserve the satisfiability of the query. To do so, *Iop* only replaces template variables with concrete values that occur in the target dataset. This requires knowledge of what the applicable instantiations are which, when repeatedly applied to a generalized query, produce a query that is satisfiable by the target dataset. One practical way to know them is to query the target dataset for them. We therefore take each query in $\mathcal{Q}^G$ and replace all the variables in the projection with all the template variables. So if a generalized query is:

```
SELECT DISTINCT ?s ?title ?pic WHERE {
  ?s  rdf:type ?ct1 ; dc:title ?title ; ?opt1 ?pic .
}
```

then the discovery query for possible instantiations is the same as the one above, except that the projection, i.e. the variables in the `SELECT` clause, becomes `?ct1 ?opt1`.

The resulting query is run through D^T. Every solution returned (the URI bindings for `ct1`, `opt1`) denotes the possible instantiations that generate a candidate query for recommendation.

The set of solutions for every such query derived from $\mathcal{Q}^G$ defines the space of candidate queries for recommendation that can be obtained through instantiation.

Every time an operation from the specialization step is applied, a new query is generated, which reduces the occurrences of template variables compared to the query from before the operation was applied. Any query generated in this way is an *intermediate query* if at least one template variable occurs, or a *candidate query* otherwise. Generalized, intermediate and candidate queries are organized in a structure that is explored in the next step. To that end, create a directed graph (digraph) $g = (V_g, E_g)$, called **specialization graph**, so that:

1. Every generalized query q_{0l} in $\mathcal{Q}^G$ is a vertex in V_g.
2. If a vertex q_{ij} is in V_g and an operation in the specialization step can be applied to it, so that a new query q_{i+1k} is generated, then $q_{i+1k} \in V_g$ and $(q_{ij}, q_{i+1k}) \in E_g$.

It follows that g contains all the generalized, intermediate and candidate queries as vertices, and that the candidate queries have no outgoing edges. Also every applicable operation on a query denotes an edge in g labelled after it. Figure 1 shows an example.

The next step is to navigate the specialization graph in order to pick the best candidate queries, trying to detect them as early as possible.

3.3 Evaluation Step

The goal now is to measure the appropriateness of a candidate query using distance measures from one of the generalized queries. With the specialization graph in place, if we set weights on its edges we will have reduced our problem to one of finding the shortest paths to traverse a digraph, from a generalized query to a candidate query. Shortest path traversal can be performed using textbook algorithms that compute the least costly paths. However, in order to assign weights to the edges, we will need to have a cost model.

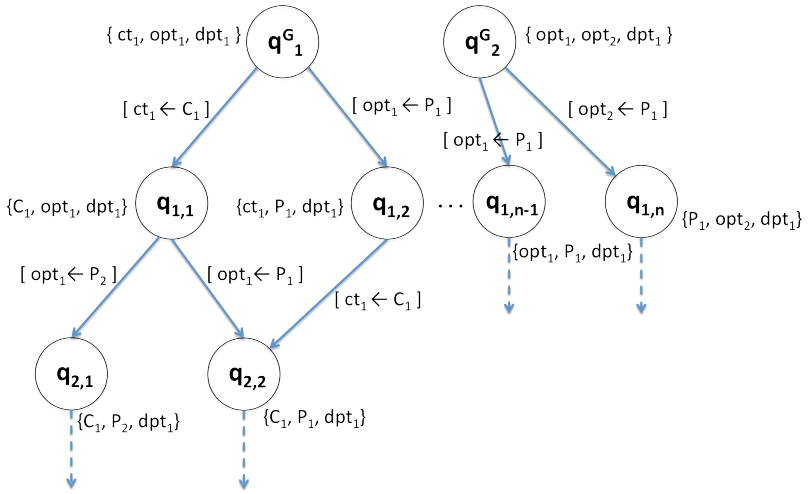


Figure 1 Example specialization graph of an lgg $\mathcal{Q}^G = \{q_1^G, q_2^G\}$ (unweighted). The edges are annotated with the last operation performed, whereas the vertices are annotated with the sets of (remaining) template variables and/or the values used to substitute them in the corresponding query.

As there is no unique way of measuring the distance and/or similarity between two queries [9], the cost C of going from q_i to a q_j satisfiable w.r.t. D is a linear combination:

$$C(q_i, q_j, D) = \alpha \cdot qsd(q_i, q_j) + \beta \cdot tpc(q_i, q_j) + \gamma \cdot ppd(q_i, q_j, D) + \delta \cdot \Phi \tag{1}$$

where $\alpha, \beta, \gamma, \delta$ are arbitrarily set coefficients that depend on how we wish to reward or penalize a certain measure, and *qsd*, *tpc* and *ppd* are measures defined as follows:

Query Specificity Distance (*qsd*) measures the alterations of variables and triple patterns:

$$qsd(q_i, q_j) = qsd^{Var}(q_i, q_j) + qsd^{TP}(q_i, q_j) \tag{2}$$

qsd^{Var} is the ratio of variables in q_i that were preserved by the operation, over those of q_i and q_j combined. qsd^{TP} does the same with the triple patterns of q_i that were preserved.

Triple Pattern Collapse (*tpc*) measures the *increase* in occurrences of every concrete value from one query to another. If $occ(x, q)$ extracts the set of occurrences $(triplepattern, role)$ of a variable or value x in a query q, then:

$$tpc(q_i, q_j) = \sum_{u \in concrete(q_i)} ||occ(u, q_j) \setminus occ(u, q_i)|| \tag{3}$$

Only Iop operations can cause an increase in value occurrence, by instantiating one or more occurrences of a template variable to an existing URI value. The side effect is that two or more TPs are *collapsed* by rendering them virtually indistinguishable. This measure imposes a penalty on queries with such TPs. Rop operations are not penalized.

Property Preservation Distance (*ppd*) counts the properties whose nature has changed across datasets, e.g. replacing an object property with a datatype property or vice versa.

Finally, Φ is a black-box similarity measure, which can be based on the syntactic or semantic analysis of the query. It is treated like a black box here, since we assume to know nothing about it: at a minimum one could use simple string similarity, just to allow us to prefer one edge over others that would otherwise be equally weighted.

We apply the cost function $C(q_{ij}, q_{i+1k}, D^T)$ to all the vertices connected by an edge and use the result as the weight of that edge. We can then proceed to visit the graph. The least costly paths, from a vertex that represents a generalized query, to each final vertex, can be found using an algorithm such as Bellmann-Ford or Dijkstra, depending on whether the coefficients $[\alpha - \delta]$ can be negative or not [2]. The resulting list of final vertices, already sorted by ascending cost to reach, constitutes the query recommendation.

4 Implementation

Our approach was implemented as part of the SQUIRE open source toolkit for SPARQL query recommendation[3]. SQUIRE provides query recommendation facilities packaged as: (a) a Java library to be included into other programs; (b) a standalone application from the command line; (c) a Web Service with an associated Web Application frontend. SQUIRE supports Lucene-based[4] dataset indexing to inform the *generalization* and *specialization* steps of the method. The indices were populated through paginated exploratory SPARQL queries.

The base implementation of SQUIRE can be extended with metrics based, for instance, on the semantic analysis of queries and datasets (i.e. the Φ in Formula (1)). As we are currently evaluating our approach, we are minimizing the bias of this factor by instead using an intentionally naïve Jaro-Winkler similarity [4] computed on the class and property labels, or on the path ends or fragments of their URIs. We are constructing a suite of queries a datasets pairs to benchmark query recommendation. At the time of writing, the optimal recommendation could be found among the first five candidates for most queries tried so far.

5 Conclusions and future directions

We have illustrated an approach to the recommendation of SPARQL queries for datasets that the user does not know, based on a query that they are already able to formulate for another dataset. Our initial study concentrated on assessing how far it is possible to go with an analysis of the query and datasets that is largely structural, only considering basic RDFS and OWL constructs, such as classes as properties, as potential invariants of such structures. We started with star-shaped queries, an essential yet already challenging structural category.

With experimental validation currently underway, we noted that there is no benchmark for SPARQL recommendation, therefore we set out to publish one to complement our evaluation. As for extending the method itself, there are several directions to take. One is to extend the structural analysis to other classes of queries, such as *snowflake* and *chain*. Another one is of course to introduce other types of analysis (such as those based on semantic alignments or subsumption hierarchy) and measure if our structural analysis aids the convergence to better recommendations sooner than without it. Having multiple analysis techniques will also allow a comparative evaluation of the method's efficiency. Finally, we will keep refining the structural analysis method itself, especially considering other types of operation for generalizing and specializing queries, and possibly taking FILTER clauses into account.

[3] SQUIRE on GitHub, `https://github.com/carloallocca/Squire`
[4] Apache Lucene, `http://lucene.apache.org`

References

1 Carlo Allocca, Alessandro Adamou, Mathieu d'Aquin, and Enrico Motta. SPARQL query recommendations by example. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenic, Sören Auer, and Christoph Lange, editors, *The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, volume 9989 of *Lecture Notes in Computer Science*, pages 128–133, 2016. doi:10.1007/978-3-319-47602-5_26.

2 Jørgen Bang-Jensen and Gregory Z. Gutin. *Digraphs - theory, algorithms and applications*. Springer, 2002.

3 Lorenz Bühmann, Jens Lehmann, and Patrick Westphal. DL-Learner - A framework for inductive learning on the Semantic Web. *J. Web Sem.*, 39:15–24, 2016. doi:10.1016/j.websem.2016.06.001.

4 William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A Comparison of String Distance Metrics for Name-Matching Tasks. In *IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03)*, pages 73–78, 2003. URL: http://www.isi.edu/info-agents/workshops/ijcai03/papers/Cohen-p.pdf.

5 G. Correndo, M. Salvadores, I. Millard, H. Glaser, and N. Shadbolt. SPARQL query rewriting for implementing data integration over linked data. In *Proceedings of the 2010 EDBT/ICDT Workshops*, EDBT '10, NY, USA, 2010. ACM. doi:10.1145/1754239.1754244.

6 Mathieu d'Aquin and Enrico Motta. Extracting Relevant Questions to an RDF Dataset Using Formal Concept Analysis. In *Proc. of the 6th K-CAP*, USA, 2011. doi:10.1145/1999676.1999698.

7 Géraud Fokou, Stéphane Jean, Allel HadjAli, and Mickaël Baron. RDF query relaxation strategies based on failure causes. In *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Proceedings*, pages 439–454, 2016. doi:10.1007/978-3-319-34129-3_27.

8 Andrey Gubichev and Thomas Neumann. Exploiting the query structure for efficient join ordering in SPARQL queries. In *17th International Conference on Extending Database Technology, EDBT 2014*, pages 439–450, 2014. doi:10.5441/002/edbt.2014.40.

9 F. Picalausa and S. Vansummeren. What Are Real SPARQL Queries Like? In *Proceedings of*, SWIM '11, pages 7:1–7:6, New York, NY, USA, 2011. ACM. doi:10.1145/1999299.1999306.

10 B. R. Kuldeep Reddy and P. Sreenivasa Kumar. Efficient approximate SPARQL querying of Web of Linked Data. In *6th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2010), collocated with ISWC-2010*, pages 37–48, 2010. URL: http://ceur-ws.org/Vol-654/paper4.pdf.

11 Umberto Straccia and Raphaël Troncy. Towards Distributed Information Retrieval in the Semantic Web: Query Reformulation Using the oMAP Framework. In *3rd European Semantic Web Conference, ESWC 2006, Proceedings*, pages 378–392, 2006. doi:10.1007/11762256_29.

12 Maria-Esther Vidal, Edna Ruckhaus, Tomas Lampo, Amadís Martínez, Javier Sierra, and Axel Polleres. Efficiently Joining Group Patterns in SPARQL Queries. In *7th Extended Semantic Web Conference, ESWC 2010, Proceedings, Part I*, pages 228–242, 2010. doi:10.1007/978-3-642-13486-9_16.

13 Gideon Zenz, Xuan Zhou, Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. From keywords to semantic queries - Incremental query construction on the Semantic Web. *J. Web Sem.*, 7(3):166–176, 2009. doi:10.1016/j.websem.2009.07.005.

OWL^C: A Contextual Two-Dimensional Web Ontology Language

Sahar Aljalbout
University of Geneva, Switzerland
sahar.aljalbout@unige.ch

Didier Buchs
University of Geneva, Switzerland
didier.buchs@unige.ch

Gilles Falquet
University of Geneva, Switzerland
gilles.falquet@unige.ch

Abstract

Representing and reasoning on contexts is an open problem in the semantic web. Despite the fact that context representation has for a long time been treated locally by semantic web practitioners, a recognized and widely accepted consensus regarding the way of encoding and particularly reasoning on contextual knowledge has not yet been reached by far. In this paper, we present OWL^C: a contextual two-dimensional web ontology language. Using the first dimension, we can reason on contexts-dependent classes, properties, and axioms and using the second dimension, we can reason on knowledge about contexts which we consider formal objects, as proposed by McCarthy [20]. We demonstrate the modeling strength and reasoning capabilities of OWL^C with a practical scenario from the digital humanity domain. We chose the Ferdinand de Saussure [15] use case in virtue of its inherent contextual nature, as well as its notable complexity which allows us to highlight many issues connected with contextual knowledge representation and reasoning.

2012 ACM Subject Classification Computing methodologies → Artificial intelligence

Keywords and phrases Contextual Reasoning, OWL^C, Contexts in digital humanities

Digital Object Identifier 10.4230/OASIcs.LDK.2019.2

Acknowledgements We would like to thanks our Saussurian colleagues in particular dr. Guiseppe Cosenza for his collaboration on FDS knowledge acquisition.

1 Introduction

The representation of context-dependant knowledge in the Semantic Web (SW) is a crucial issue. Several paradigms have been proposed with the aim of adding context awareness into the SW; ranging from practical RDF graph design patterns [23] [13] to theoretical works on extending description logic languages with contextual constructs and axioms [5] [18]. In this work, we present a novel approach as a combination of a formally defined theory and a practical implementation of contextual reasoning with OWL.

Before starting, let's clarify what do we mean by contexts and contextual reasoning. We consider that triples can be enriched with two-types of contexts: i) validity contexts which enhance the meaning of a fact such as the temporal validity. The fact itself is not sufficiently clear without validity contexts ii) additional contexts which add to the fact without interfering with its meaning such as the provenance of the triple. A statement where both contexts are given is the following: *Saussure lived in Geneva between 1857 and 1876 as mentioned by Wikipedia*, where 1857-1876 represents the validity context (more precisely the validity time) and Wikipedia is the provenance considered as an additional context. Based on that, we define contextual reasoning as the process of deriving new contextual knowledge from existing ones. The kernel of this process is reasoning on contexts themselves in order to boost the propagation of contextual knowledge.


© Sahar Aljalbout, Didier Buchs, and Gilles Falquet;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 2; pp. 2:1–2:13

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

For all this, we propose OWL^C, a contextual two-dimensional web ontology language that is an extension of the classical OWL. OWL^C [1] [4] was designed in the two-dimensions style [17] in the purpose of 1) avoiding the conflict when modeling contexts and context-dependent knowledge 2) avoiding adding an additional cost in the complexity of reasoning because the cost is already hidden in the shift from one-dimensional to two-dimensional semantics. Furthermore, the design of OWL^C was inspired by problems we encountered in practical scenarios in digital humanities. Therefore, we chose to test its usability over the SNSF[1] project of Ferdinand de Saussure [3], which is sufficiently complex and paradigmatic to contain different aspects of context-dependent knowledge.

The remainder of the paper has been organized as follows: in section 2, we present the Ferdinand De Saussure (FDS) use case. In section 3, we go through the literature review of contextual knowledge representation and reasoning. In section 4 and 5, we present a contextual extension of OWL: OWL^C. We discuss also the different types of reasoning that can be performed. Furthermore, we demonstrate the usability of OWL^C by applying it to a historical scenario in section 6. Finally, we summarize our results in section 7.

2 Motivation: the Case of Ferdinand de Saussure (FDS)

Ferdinand de Saussure (1857 – 1913) is considered as a "formidable linguist" [15], first of all for his works in general linguistics, as well for his contributions in the rather more exclusive field of comparative grammar. However, Saussure published very little. For instance, he never published the theory he developed in the course of general linguistics he taught three times and which is considered as the work of his life. It is on the basis of lecture notes of his students that the book Course in General Linguistics (Cours de Linguistique générale CLG) was published in 1916. The legacy of Saussure is fortunately not limited to these monographs but includes a fund of about 50,000 handwritten pages[2] deposited in libraries of Geneva (Bibliothèque de Genève), Paris and Harvard. All these pages were photographed using a high definition digital camera. These manuscripts are of primordial importance for the Saussurean scholars (Saussureans for short). Their study is considered as the only mean to reach a better understanding of Saussure's ideas. As of today, only 5,000 manuscripts of the 50'000 pages have been transcribed. One of the major problems of Saussureans is to understand the content of the manuscripts and this is due to the following contextual problem:

- Authorship as a context: transcripts of manuscripts come from various sources. Their authorship is of major importance for Saussurians given the level of confidence that they attribute to each source.
- Time as a context: for the majority of the manuscripts, we know neither their date nor their place of writing. This, of course, complicates the establishment of a clear sequence of ideas on Saussure's work.
- Terminology as a context: in [10], the author showed that the terminology used by Saussure varies over time or writing purpose. He eventually identified more than a dozen different terminologies in Saussure's work. Therefore, the terminology can also be considered as a context. Indeed it is essential to precisely understand the meaning of each specific manuscript.

[1] `http://www.snf.ch/en/Pages/default.aspx`
[2] which have been (and still) transcribed

3 Related Works

In 2001, the authors of [12] introduced the idea of locality and compatibility where reasoning is considered mainly local and uses only part of what is potentially available. In 2003, [7] introduced the concept of distributed description logics where binary relations describe the correspondences between contexts. However, the coordination between a pair of ontologies can only happen with the use of bridge rules. C-OWL [8] was introduced in the same year. The idea behind is to localize the content of ontologies and to allow for explicit mappings via bridge rules. In 2004, a new concept called E-connections [18] emerged: ontologies are interconnected by defining new links between individuals belonging to distinct ontologies. One major disadvantage is that it does not allow concepts to be subsumed by concepts of another ontology, which limits the expressiveness of the language. Then, in 2006, the authors of [5] attempted to extend description logics with new constructs with relative success. In 2011, a proposition was argued to use a two dimensional-description logics [17]. Results showed that this approach does not necessarily increase the computational complexity of reasoning. Another work, [16], proposed a framework for contextual knowledge representation and reasoning (CKR) based on current RDF(S) standards. However, the expressiveness of the formalism is restricted to RDFS and there are no axioms that make it possible to explicitly use the relationships between contexts to deduce new facts or to deal with contradictions between contexts. In 2012, [9] argues that treating contexts in the semantic web needs more advanced means, such that contexts should be explicitly presented and logically treated...

On the other hand, many attempts to find a solution to the syntactic restriction of RDF binary relations emerged. Two approaches were proposed:

(a) Extending the data model and/or the semantics of RDF: the triple data structure could be extended by adding a fourth element to each triple, which is intended to express the context [11] of a set of triples [14] [21].

(b) Using design patterns: It could be categorized along three axes:
- the contextual index co is attached to the statement R(a,b) and thus R(a,b) holds for co such as RDF reification [6]. This method is not supported in DL reasoning.
- the contextual index co is attached to the relation R(a,b,co) [2] [3]. One advantage is being able to talk about assertions as (reifying) individuals.
- the contextual index co is attached to the object terms R(a@co, b@co) where co is the contextual-slice of a and b [22]. This method introduces many contextualized individuals which cause objects proliferation.

4 OWL 2 DL^C: a Two-dimensional Web Ontology Language for Contexts

OWL 2 DL was designed to support the existing description logic business segment and has desirable computational properties for reasoning systems. In this section, we introduce an extension of OWL 2 DL for contexts, that we call OWL 2 DL^C. The semantics are based on the semantics of the two-dimensional description logic [17]. OWL 2 DL$^C_{core}$ is the first dimension. It is used to represent contextual object knowledge such as contextual classes, contextual properties and contextual axioms. OWL 2 DL$^C_{context}$ is the second dimension. It is used to represent contexts which are considered as first class citizens.

Formally speaking, an OWL 2 DL^C signature (or vocabulary) is a pair of DL signatures $(\langle N_C, N_R, N_I\rangle, \langle N_{KC}, N_{KR}, N_{KI}\rangle)$ where:
- N_C (resp. N_{KC}) is a set of domain (resp. context) concept names,
- N_R (N_{KR}) is a set of domain (context) role names,
- N_I (N_{KI}) is a set of domain (context) individuals names.

4.1 The contexts language: OWL 2 DL$^C_{context}$

Contexts are considered as formal objects [20] and are of two types:

- Validity contexts: are contexts that can affect the fact itself either by enhancing its meaning, or by limiting its meaning to a given context. Fluents [23] are a typical example of validity contexts (i.e. a fluent is a temporal property whose object is subject to change over time).
- Additional contexts: supplement a fact with additional elements that do not modify its meaning. As a result, the fact is more precisely described with the additional context, but sufficiently clear without it. A typical example is the publication context which provide information about the provenance of the triple as a reference in order to support the claim.

A context type is usually characterized by a set of dimensions that describe it to a certain level of approximation. For instance, a validity context could be composed of many dimensions, such as the temporal validity, the spatial validity, etc. For example:

$$(1857, wikipedia) : LivedIn(Saussure, Geneva)$$

states that Saussure lived in Geneva during 1857 as mentioned in Wikipedia. 1857 is the temporal dimension of the validity context and Wikipedia is the provenance dimension considered as an additional context[3].
The axioms of the contexts language are formulas:

$$A \sqsubseteq B \mid C(a)$$

where A $\in N_{KC}$, B $\in N_{KC}$, C $\in N_{KC}$, a $\in N_{KI}$.

4.2 The core language: OWL 2 DL$^C_{core}$

An axiom expression of the core language is either:

- a DL axiom expression on the core signature $\langle N_C, N_R, N_I \rangle$. For Example:

$$Human(Saussure)^4$$

- an expression of the form $K : \phi$, where K is either an individual context name (in N_{KI}) or a concept expression over the context signature $\langle N_{KC}, N_{KR}, N_{KI} \rangle$. Such an expression states that the axiom ϕ holds in the specified context or in all contexts of the specified context concept. ϕ can be:
  1. a concept axiom ($C \sqsubseteq D$, $C \equiv D$, C *disjoint* D)

$$1969 : CanVote \sqsubseteq Aged21orMore$$

states that the axiom $CanVote \sqsubseteq Aged21orMore$ holds in the temporal context *1969*.

[3] In this case the individual context names N_{KI} is the cartesion product $N_{KIt} \times N_{KIp}$ of a set of temporal contexts and a set of provenance contexts.

[4] We consider non contextual (standard) DL axioms as contextual axioms that are valid in all validity contexts. Therefore an expression of the form , $C \sqsubseteq D$ is in fact an abbreviation for $\top^{VC} : C \sqsubseteq D$ where $\top^{VC}$ is the top context concept whose interpretation contains all the validity contexts of Ω.

Table 1 OWL 2 DL^{C}_{core} direct model theoretic semantics.

Abstract Syntax	CDL syntax	Semantics (Interpretation in context k)
IntersectionOf(C_1 ... C_n)	$C_1 \sqcap ... \sqcap C_n$	$C_1^{\mathcal{I}[k]} \cap ... \cap C_n^{\mathcal{I}[k]}$
UnionOf(C_1 ... C_n))	$C_1 \sqcup ... \sqcup C_n$	$C_1^{\mathcal{I}[k]} \cup ... \cup C_n^{\mathcal{I}[k]}$
ComplementOf(C)	$\neg C$	$(\neg C)^{\mathcal{I}[k]} = \Delta^{\mathcal{I}[k]} \setminus C^{I[k]}$
R SomeValuesFrom(C)	$\exists(R.C)$	$x \vert \exists y : (x,y) \in (R)^{\mathcal{I}[k]} and \quad y \in (C)^{\mathcal{I}[k]}$
R AllValuesFrom(C)	$\forall(R.C)$	$x \vert \forall y : (x,y) \in (R)^{\mathcal{I}[k]} \rightarrow y \in (C)^{\mathcal{I}[k]}$
OneOf(a_1 ...a_n)	a_1 ... a_n	$(a_1)^{\mathcal{I}[k]}, ..., (a_n)^{\mathcal{I}[k]}$

2. a role axiom ($R \sqsubseteq S$, $functional(R)$, $transitive(R)$, ...)

   $$DecentralizedCountry : hasLocalPowerIn \sqsubseteq electedLocallyIn$$

   states that in decentralized countries (contexts), a person with local power in a region had necessarily been locally elected in that region.
3. a class or role assertion ($C(a)$, $R(a,b)$) defined on the core signature with contextual concept and role expressions

   $$1857 : Professor(Saussure)$$

   which states that Saussure was a professor during 1857.

A contextual interpretation is a pair of interpretations $\mathcal{M} = (\mathcal{I}, \mathcal{J})$ where $\mathcal{I} = (\Delta, .^{\mathcal{I}[.]})$ is the core interpretation, $\mathcal{J} = (\Omega, .^{\mathcal{J}})$ is the context interpretation, and $\Delta \cap \Omega = \emptyset$. $.^{\mathcal{I}[.]}$ is a family of interpretation functions, one for each context $k \in \Omega$. $.^{\mathcal{J}}$ is the (non-contextual) interpretation function of every context in the context language. The interpretation of the class constructors of the core language is straightforward. Table 1 contains the OWL-frame like abstract syntax, the contextual description logic syntax (CDL) and the direct model theoretic semantics of OWL^{C}_{core} basic class constructors. We only consider contextual interpretations that satisfy the *rigid designator hypothesis* [19], i.e. $i^{\mathcal{I}[k]} = i^{\mathcal{I}[k']}$ for any individual $i \in N_I$, $k \in \Omega$, and $k' \in \Omega$.

A contextual axiom $K : \phi$ is satisfied by an interpretation $\mathcal{M}$ if in every context k that belongs to the interpretation of K, the interpretation in k of the concepts, roles and individuals that appear in ϕ satisfy the axiom condition

- $\mathcal{M} \models K : C \sqsubseteq D$ iff $\forall k \in K^{\mathcal{J}} : C^{\mathcal{I}[k]} \subseteq D^{\mathcal{I}[k]}$, where $C \in N_C$ and $D \in N_C$
- $\mathcal{M} \models K : R \sqsubseteq S$ iff $\forall k \in K^{\mathcal{J}} : R^{\mathcal{I}[k]} \subseteq S^{\mathcal{I}[k]}$, where $R \in N_R$ and $S \in N_R$
- $M \models K : C(a)$ iff $\forall k \in K^{\mathcal{J}} : C(a)^{\mathcal{I}(k)}$, where $C \in N_C$ and $a \in N_I$
- $M \models K : R(a,b)$ iff $\forall k \in K^{\mathcal{J}} : R(a,b)^{\mathcal{I}(k)}$, where $R \in N_R$, $a \in N_I$ and $b \in N_I$

(if K is not a concept expression but a context individual name k, $K^{\mathcal{J}}$ designates the singleton $\{k^{\mathcal{J}}\}$ in the above expressions).

4.3 The interaction between the core and context language

The interaction between the two languages is done using special operators. We introduce, in table 2, the OWL frame-like abstract syntax and the semantics of these contexts-based concept forming operators. Examples:

- $\langle AsianCountry \rangle\, Professor$: the individuals that belong to the class $Professor$ in some context of type $xAsianCountry$.

- $[EuropeanCountry]Professor$: the individuals that belong to the class $Professor$ in all contexts of type $EuropeanCountry$.
- $\{Switzerland\}Professor$: the individuals that belong to the class $Professor$ in *Switzerland*.

Table 2 Semantics of the contexts-based concept forming operators.

Abstract Syntax	CDL	Semantics
ConceptValuesFromSomeContext(C [K])	$\langle K \rangle\, C$	$x \in \Delta \mid \exists y \in K^{\mathcal{J}} : x \in C^{\mathcal{I}[y]}$
ConceptValuesFromAllContext(C [K])	$[K]C$	$x \in \Delta \mid \forall y \in K^{\mathcal{J}} \rightarrow x \in C^{\mathcal{I}[y]}$
ConceptValuesFromThisContext(C [k])	$\{k\}\, C$	$x \in \Delta \mid x \in C^{\mathcal{I}[k^{\mathcal{J}}]}$
PropertyValuesFromSomeContext(R [K])	$\langle K \rangle\, R$	$(x,z) \in \Delta \times \Delta \mid \exists y \in K^{\mathcal{J}} : (x,z) \in R^{\mathcal{I}[y]}$
PropertyValuesFromAllContext(R [K])	$[K]R$	$(x,z) \in \Delta \times \Delta \mid \forall y \in K^{\mathcal{J}} : (x,z) \in R^{\mathcal{I}[y]}$
PropertyValuesFromThisContext(R [k])	$\{k\}\, R$	$(x,z) \in \Delta \times \Delta \mid (x,z) \in R^{\mathcal{I}[k^{\mathcal{J}}]}$

5 Reasoning with OWL^C

Inspired from OWL 2 RL[5], OWL^C is considered as a profile aimed at applications that require scalable reasoning without sacrificing too much expressive power. This is achieved by restricting the use of constructs to a certain syntactic position, similarly to OWL 2 RL.

In the original version of OWL-2 RL, the rules are given as universally quantified first-order implications over a ternary predicate T. This predicate represents a generalization of RDF triples thus, *T(s,p,o)* represents a generalized RDF triple with the subject s, predicate p, and the object o. Variables in the implications are preceded with a question mark. To include the notion of contexts, we introduce a quaternary predicate $Q(s,p,o,k)$ where s is the subject, p is the predicate, o is the object and k is the context for which the predicate holds. If the ontology has multiple context dimensions (e.g. time and provenance) k must be understood as $k_1, \ldots, k_m$ and hence Q as an $m+3$-ary predicate.

We can distinguish two types of object reasoning: explicit and implicit.

Implicit contextual reasoning

When the TBox axioms is declared as in normal OWL but the ABox is contextual.

$Professor \sqsubseteq hasColleague$ **only** $Professor$
$1904 : Professor(Ferdinand)$
$1904 : hasColleague(Ferdinand, Robert)$
$1880 : hasColleague(Ferdinand, Clara)$
entails $1904 : Professor(Robert)$ but not $1880 : Professor(Clara)$.

Explicit contextual reasoning

When the TBox axioms explicitly refer to contexts. From
$FranceBefore1944 : CanVote \sqsubseteq Man$

[5] https://www.w3.org/TR/owl2-profiles/#Feature_Overview_3

Table 3 OWL^C: Entailment rules for the core language.

	IF	**THEN**
cls-com $\neg C$	T(?c_1, owl:complementOf, ?c_2) Q(?x, rdf:type, ?c_1, ?k) Q(?x, rdf:type, ?c_2, ?k)	false
cls-int1 $C \sqcap D$	T(?c, owl:intersectionOf, ?x) LIST[?x, ?c_1, ..., ?c_n] Q(?y, rdf:type, ?c_1, ?k) Q(?y, rdf:type, ?c_2, ?k) ... Q(?y, rdf:type, ?c_n, ?k)	Q(?y, rdf:type, ?c, ?k)
cls-int2 $C \sqcap D$	T(?c, owl:intersectionOf, ?x) LIST[?x, ?c_1, ..., ?c_n] Q(?y, rdf:type, ?c, ?k)	Q(?y, rdf:type, ?c_1, ?k) Q(?y, rdf:type, ?c_2, ?k) ... Q(?y, rdf:type, ?c_n, ?k)
cls-uni $C \sqcup D$	T(?c, owl:unionOf, ?x) LIST[?x, ?c_1, ..., ?c_n] Q(?y, rdf:type, ?c_i, ?k)	Q(?y, rdf:type, ?c, ?k)
cls-svf1-1 $\exists R.C$	T(?x, owl:someValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, ?p, ?v, ?k) Q(?v, rdf:type, ?y, ?k)	Q(?u, rdf:type, ?x, ?k)
cls-svf1-2 $\exists R.C$	T(?x, owl:someValuesFrom, ?y) T(?x, owl:onProperty, ?p) T(?u, ?p, ?v) Q(?v, rdf:type, ?y, ?k)	Q(?u, rdf:type, ?x, ?k)
cls-svf1-3 $\exists R.C$	T(?x, owl:someValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, ?p, ?v, ?k) T(?v, rdf:type, ?y)	Q(?u, rdf:type, ?x, ?k)
cls-avf-1 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?k) Q(?u, ?p, ?v, ?k)	Q(?v, rdf:type, ?y, ?k)
cls-avf-2 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?k) T(?u, ?p, ?v)	Q(?v, rdf:type, ?y, ?k)
cls-avf-3 $\forall R.C$	T(?x, owl:allValuesFrom, ?y) T(?x, owl:onProperty, ?p) Q(?u, rdf:type, ?x, ?k) Q(?u, ?p, ?v, ?k)	T(?v, rdf:type, ?y)

Table 4 OWL^C : entailment rules for the context-based concept forming operators.

	IF	THEN
cxt-svf ($\langle K \rangle\, D$)	T(?e, owl^c : onClass, ?d) T(?e, owl^c : inSomeContextOf, ?k) Q(?x, rdf:type, ?d, ?y) T(?y, rdf:type, ?k)	T(?x, rdf:type, ?e)
cxt-avf ($[K]D$)	T(?e, owl^c : onClass, ?d) T(?e, owl^c : inAllContextOf, ?k) T(?x, rdf:type, ?e) Q(?x, rdf:type, ?d, ?y)	T(?y, rdf:type, ?k)
cxt-ov ($\{K\}\, D$)	T(?e, owl^c : onClass, ?d) T(?e, owl^c : inThisContext, ?k) Q(?x, rdf:type, ?e)	Q(?x, rdf:type, ?d, ?k)

$FranceBefore1944 : CanVote(Alejandro)$
$FranceIn1989 : CanVote(Andros)$
we can infer $FranceBefore1944 : Man(Alejandro)$ but not $FranceIn1989 : Man(Andros)$ (where $FranceBefore1944$ and $FranceIn1989$ are the contexts in use).

Interaction between OWL$^C_{core}$ and OWL$^C_{context}$

The rules presented in this section let us do the interaction between the two languages. Syntactic restrictions are applied to the new constructors: an existential contextual restriction ($\langle C \rangle\, D$, $\langle C \rangle\, R$) may only appear in the left-hand side of a subclass axiom, whereas a universal contextual restriction ($[C]D$, $[C]R$) may only appear in the right-hand side. Due to space limitations, we show only some of these rules in table 4.

An example of the existential rule is as follows: a former president is someone who has been president in the past
$\langle PastPresidentialTerm \rangle President \sqsubseteq FormerPresident$
1933-1945 : $President(Roosvelt)$
$PastPresidentialTerm$(1933-1944)
entails $FormerPresident(Roosvelt)$

6 OWL^C in practice

Since OWL^C was created to deal with practical problems, we chose to evaluate it on a real use case: the SNSF project of Ferdinand de Saussure (FDS). Therefore, in this section, we explain the methodology to follow from the choice of contexts to reasoning. First, we start by defining the contexts dimensions to be used. In addition, we describe the process we followed to extract contextual knowledge from the Saussurian texts. Then, we discuss the problems we encountered while encoding the model in RDF. Finally, we propose a practical implementation of contextual reasoning.

6.1 How to choose your context dimensions?

When talking about the implementation of contextual reasoning, some questions always arise such as: how do you decide what should be a context and what shouldn't? Is there a list of predefined contexts dimensions, you choose from? etc. According to your target, you choose your dimensions. In the case of the FDS project, we are interested in reasoning about time and provenance. Therefore, we choose the validity time and provenance as our dimensions.

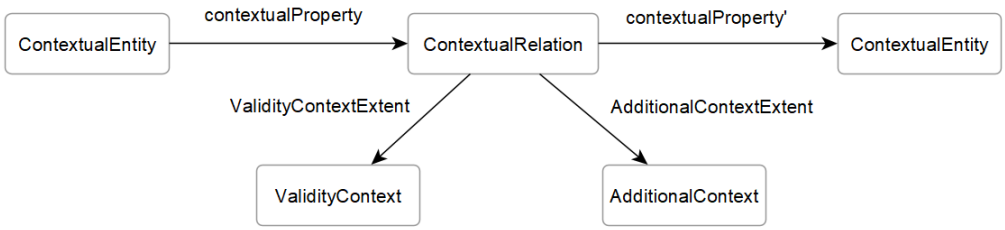


Figure 1 Contexts in RDF.

In order to come up with a suitable range of dimensional values, we must consider the granularity of contexts. In our use case, the main focus is on the Saussurian network (persons he cites in his manuscripts, students, etc.) and events he participated too. Therefore, the data provenance will be the transcriptions from which the data was extracted. For the time dimension, the most granular value is a "year".

6.2 Contextual knowledge acquisition

The acquisition of contextual knowledge was the hardest phase of this project given the fact that: 1) the information is scattered in thousands of transcriptions 2) no general purpose natural language processing tool can extract accurately knowledge from text yet, in particular, contextual entities or more precisely n-ary relations (e.g. Saussure lived in Geneva in 1857). In many cases, information could be split over different sentences, so the problem can be hard and require "coreference resolution". The simplest way was to use existing tools to find binary relations and then parse in the vicinity of the text to find contexts such as dates/years. In cooperation with a Saussurian linguist, we did the task semi-automatically. Using Gate[6], we extracted name entities and relations from transcriptions. Time and provenance were then added to the contextual relations. Knowledge was also enriched with Wikidata[7]. We have 1032 persons. We have also shown in [2] that the FDS project contains a lot of fluent[8] relations among them: relations between persons (colleagues, studentOf, professor, spouseOf, husbandOf, educatedAt, etc.).

6.3 Representing FDS with OWL^C

In this section, we explain how to encode the overall model in RDF. We start by presenting the contextual pattern we adopted and then we prove the correspondence between the OWL^C formalization and the RDF based representation.

When it comes to encoding contexts in RDF, a lot of techniques are made available (check section 3). We chose to use the n-ary pattern we presented in [2] for its compactness and intuitiveness (figure 1). In order to map OWL^C to RDF, we implicitly used the standard

[6] `https://gate.ac.uk/projects.html`
[7] `https://www.wikidata.org/wiki/Wikidata`
[8] a fluent is a relation whose object is subject to change over time (e.g Saussure lives in Geneva in 1860 but in Paris in 1882)

mapping of OWL to RDF[9]. For instance, the mapping of the axiom:

$$(1904, UniversityOfGeneva) : Colleague(Saussure, Paolo)$$

where $(1904, UniversityOfGeneva)$ is the validity context composed of the validity time (1904) and the location ($UniversityOfGeneva$), is as follows:

```
:Saussure   cp:colleagueOf          :x.
:x          cp1:colleagueOf         :Ascoli.
:x          rdf:type                :contextualRelation.
:x          :during                 :"1889"^^xsd:date.
:x          :location               :UniversityOfGeneva.
:during     rdfs:subPropertyOf      owlc:validityContextExtent
:location   rdfs:subPropertyOf      owlc:validityContextExtent
```

Where

- `cp` is used for the property linking the entity to the contextual relation.
- `cp1` is used for the property linking the contextual relation to the object.
- `owlc` refers to the vocabulary introduced by the contextual ontology.

The mapping of the context-based concept forming operators to RDF is more delicate. In order to represent the contextual existential $\langle C \rangle\, D$ and universal operators [C]D, we designed the *owlc:contextRestriction* similarly to *owl:Restriction*. A context restriction class should have exactly two triples linking the restriction to:

1. the class (resp. property) that the restriction applies on, using the new predicate *owlc:onClass* (*owl:onProperty*)
2. The type of the restriction: in case of a universal (resp. existential) restriction , *owlc:inAllContextOf* (*owlc:inSomeContextOf*) should be used.

If

$$[EuropeanCountries]FamousLinguist$$

represents the people who are considered as famous linguists in all european countries. The mapping is as follows:

```
_:x    rdf:type              owlc:ContextRestriction .
_:x    owlc:onClass          :FamousLinguist .
_:x    owlc:inAllContextOf   :EuropeanCountries.
```

6.4 Reasoning in FDS with OWL^C

One characteristics of the contextual rules is that they generates new objects of type ContextualRelation. We choose to use SPIN[10][4] because it is flexible enough that you can pass parameters to them to customize their behavior. Then, they can be instantiated in any RDF or OWL ontology to add inference rules and constraint checks. Two types of rules were implemented using TopBraid Composer[11]:

[9] https://www.w3.org/TR/owl2-mapping-to-rdf/
[4] http://spinrdf.org
[11] https://www.topquadrant.com/tools/ide-topbraid-composer-maestro-edition/

6.4.1 OWL^C rules

Figure 2 shows the example of the *cls-int* rule encoded as a SPIN template. It declares that the assertion of the same individual in two classes, holding for the same context, generates an assertion for this individual in the intersection of those classes, but also for the same holding contexts. It is implemented using a SPARQL INSERT request and is composed of a *spin* : *body* and *spin* : *constraint*.

spin:body

```
INSERT{
?this   owlc:representedBy          _:b0.
_:b0    a                           owlc:ContextualRelation.
_:b0    a                           ?ClassIntersection.
_:b0    owlc:validityContextExtent  ?co.
 }
WHERE{
?this   owlc:representedBy          ?cr1.
?cr1    a                           owlc:ContextualRelation.
?cr1    a                           ?FirstClass.
?cr1    owlc:validityContextExtent  ?co.
?this   owlc:representedBy          ?cr2.
?cr2    a                           owlc:ContextualRelation.
?cr2    a                           ?SecondClass.
?cr2    owlc:validityContextExtent  ?co.
    FILTER NOT EXISTS{
    ?this   owlc:representedBy          _:0.
    _:0     a                           owlc:ContextualRelation.
    _:0     a                           ?ClassIntersection.
    _:0     owlc:validityContextExtent  ?co.
    }
}
```

spin:constraint

```
Argument arg:ClassIntersection   rdfs:Class
Argument arg:FirstClass          rdfs:Class
Argument arg:SecondClass         rdfs:Class
```

Notice that the classes are declared as *spin:constraint*. Notice also that the query contains a filter. The existence of the filter is of a major importance, because it guarantees that an existing triple is not generated again and again, whenever the rules are running.

6.4.2 Domain rules

Domain rules where added to enable historical reasoning over the knowledge. They were created in collaboration with Saussurean experts. A typical rule is

> If a manuscript M is a letter written by a scholar A to a scholar B at time t then we can infer that A is aware of B's work at time t and thereafter, i.e in the time interval $[t, end\ ofconsidered\ period]$.

For instance, from the fact that a manuscript M is written by a person A as a letter to a person B and the writing time of M is $[t1...t2]$, we can infer that A knows B since $t1$.

7 Conclusion

OWL^C is an extension of the web ontology language for contexts. It is completely embedded within the current Semantic Web standards. It builds on top of these standard formalisms and enhances them with the following aspects: (1) knowledge is organized in two layers: contextualized knowledge and knowledge about contexts (2) contexts can have many dimensions and are divided into validity context and additional context (3) reasoning can be performed explicitly or implicitly. We also described a modeling scenario from the domain of digital humanities, by which we demonstrate the features of OWL^C. The choice of this particular domain is due to its inherent contextual nature and sufficient complexity.

References

1 Sahar Aljalbout, Didier Buchs, and Gilles Falquet. Introducing Contextual Reasoning to the Semantic Web with OWL^C. In *Proceedings of the 24th International Conference on Conceptual Structures: ICCS*. Springer, 2019.

2 Sahar Aljalbout and Gilles Falquet. Un modèle pour la représentation des connaissances temporelles dans les documents historiques. In *IC 2017 : 28es Journées francophones d'Ingénierie des Connaissances (Proceedings of the 28th French Knowledge Engineering Conference), Caen, France, July 3-7, 2017.*, pages 145–150, 2017. URL: `https://hal.archives-ouvertes.fr/hal-01568018`.

3 Sahar Aljalbout and Gilles Falquet. A Semantic Model for Historical Manuscripts. *arXiv preprint*, 2018. `arXiv:1802.00295`.

4 Sahar Aljalbout, Gilles Falquet, and Didier Buchs. A Practical Implementation of Contextual Reasoning on the Semantic Web. In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 2: KEOD,*, pages 255–262. INSTICC, SciTePress, 2018.

5 Djamal Benslimane, Ahmed Arara, Gilles Falquet, Zakaria Maamar, Philippe Thiran, and Faiez Gargouri. Contextual ontologies. *Advances in Information Systems*, pages 168–176, 2006.

6 Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

7 Alexander Borgida and Luciano Serafini. Distributed description logics: Assimilating information from peer sources. *J. Data Semantics*, 1:153–184, 2003.

8 Paolo Bouquet, Fausto Giunchiglia, Frank Van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. C-owl: Contextualizing ontologies. In *International Semantic Web Conference*, pages 164–179. Springer, 2003.

9 Loris Bozzato, Martin Homola, and Luciano Serafini. Context on the semantic web: Why and how. *ARCOE-12*, page 11, 2012.

10 Giuseppe Cosenza. *Tra terminologia e lessico: i percorsi di pensiero di F. de Saussure*. PhD thesis, University of Calabria, 2015.

11 Renata Dividino, Sergej Sizov, Steffen Staab, and Bernhard Schueler. Querying for provenance, trust, uncertainty and other meta knowledge in RDF. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):204–219, 2009.

12 Chiara Ghidini and Fausto Giunchiglia. Local Models Semantics, or contextual reasoning=locality+compatibility. *Artificial intelligence*, 127(2):221–259, 2001.

13 José M Giménez-García, Antoine Zimmermann, and Pierre Maret. NdFluents: An Ontology for Annotated Statements with Inference Preservation. In *European Semantic Web Conference*, pages 638–654. Springer, 2017.

14 Olaf Hartig and Bryan Thompson. Foundations of an alternative approach to reification in RDF. *arXiv preprint*, 2014. `arXiv:1406.3399`.

15 John E Joseph. *Saussure*. Oxford University Press, 2012.

16 Mathew Joseph and Luciano Serafini. Simple Reasoning for Contextualized RDF Knowledge. In *WoMO*, pages 79–93, 2011.

17 Szymon Klarman and Víctor Gutiérrez-Basulto. Two-Dimensional Description Logics for Context-Based Semantic Interoperability. In *AAAI*, 2011.

18 Oliver Kutz, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. E-connections of abstract description systems. *Artificial intelligence*, 156(1):1–73, 2004.

19 Joseph LaPorte. Rigid designators. *Philosophical Studies*, 130:321—336, 2006.

20 John McCarthy. Generality in artificial intelligence. *Communications of the ACM*, 30(12):1030–1035, 1987.

21 Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. Don't like RDF reification?: making statements about statements using singleton property. In *Proceedings of the 23rd international conference on World wide web*, pages 759–770. ACM, 2014.

22 Chris Welty. Context slices: representing contexts in OWL. In *Proceedings of the 2nd International Conference on Ontology Patterns-Volume 671*, pages 59–60. CEUR-WS. org, 2010.

23 Chris Welty, Richard Fikes, and Selene Makarios. A reusable ontology for fluents in OWL. In *FOIS*, volume 150, pages 226–236, 2006.

Ligt: An LLOD-Native Vocabulary for Representing Interlinear Glossed Text as RDF

Christian Chiarcos
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
http://www.acoli.informatik.uni-frankfurt.de/
chiarcos@informatik.uni-frankfurt.de

Maxim Ionov
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
ionov@informatik.uni-frankfurt.de

Abstract

The paper introduces Ligt, a native RDF vocabulary for representing linguistic examples as text with interlinear glosses (IGT) in a linked data formalism. Interlinear glossing is a notation used in various fields of linguistics to provide readers with a way to understand linguistic phenomena and to provide corpus data when documenting endangered languages. This data is usually provided with morpheme-by-morpheme correspondence which is not supported by any established vocabularies for representing linguistic corpora or automated annotations.

Interlinear Glossed Text can be stored and exchanged in several formats specifically designed for the purpose, but these differ in their designs and concepts, and they are tied to particular tools, so the reusability of the annotated data is limited. To improve interoperability and reusability, we propose to convert such glosses to a tool-independent representation well-suited for the Web of Data, i.e., a representation in RDF. Beyond establishing structural (format) interoperability by means of a common data representation, our approach also allows using shared vocabularies and terminology repositories available from the (Linguistic) Linked Open Data cloud.

We describe the core vocabulary and the converters that use this vocabulary to convert IGT in a format of various widely-used tools into RDF. Ultimately, a Linked Data representation will facilitate the accessibility of language data from less-resourced language varieties within the (Linguistic) Linked Open Data cloud, as well as enable novel ways to access and integrate this information with (L)LOD dictionary data and other types of lexical-semantic resources. In a longer perspective, data currently only available through these formats will become more visible and reusable and contribute to the development of a truly multilingual (semantic) web.

2012 ACM Subject Classification Information systems → Graph-based database models; Computing methodologies → Language resources; Computing methodologies → Knowledge representation and reasoning

Keywords and phrases Linguistic Linked Open Data (LLOD), less-resourced languages in the (multilingual) Semantic Web, interlinear glossed text (IGT), data modeling

Digital Object Identifier 10.4230/OASIcs.LDK.2019.3

Funding The research described in this paper was conducted in the project *Linked Open Dictionaries* (LiODi, 2015-2020), funded by the German Ministry for Education and Research (BMBF) as an Early Career Research Group on eHumanities.

1 Background

Interlinear glossed text (IGT) is a notation frequently used in linguistic research and documentation. IGTs combine language utterances with their morphological analysis in order to provide readers with a way to understand linguistic phenomena in languages they do not necessarily know. An important property of IGT examples is that there is an alignment


© Christian Chiarcos and Maxim Ionov;
licensed under Creative Commons License CC-BY

between morphemes and corresponding grammatical values as in (1)[1]:

(1) Min ter-a ter-gan ezba tau-da bas-kan
I live-ST.IPFV live-PFCT house hill-LOC get_up-PFCT
"I live in the house that is located on the hill" (Tatar, Mishar dialect)

Accessing and analyzing IGT data is vital for a vast amount of linguistic research, especially when dealing with less-resourced languages. However, unlike other types of linguistic resources like corpora or dictionaries, this type of data lacks interoperability and reusability. In practice, linguists rarely use IGTs from outside their research groups. There are two main reasons for this: a conceptual one and a technical one.

From a conceptual point of view, IGT can vary greatly from a researcher to researcher. The Leipzig Glossing Rules [3] define guidelines and best practices for writing glossed examples and texts, but these represent only the basis on which researchers can later build on, introducing more and refined information in their analysis (e.g., a layer with a phonetic transcription or more specific abbreviations for linguistic categories). Another source of variability lies in the list of grammatical categories. In the language documentation community, there is no single inventory for grammatical categories all over the world and their corresponding abbreviations (tags), also, there can be several ways for representing the same category (cf. `AOR` vs. `aor` vs. `aorist`). All these factors make it more difficult to redistribute or reuse IGT data.

From a technical point of view, there is another problem: There is a myriad of ways to encode IGT, each with its advantages and limitations, ranging from printed PDFs to various XML (and pre-XML) formats defined by specific tools such as Toolbox and FLEx. Even in the simple cases it can be difficult to use independently produced IGTs, and it is even more problematic to compare or combine several data sources. One possible way to overcome this problem would be to represent data in a tool- and theory-agnostic format. Several initiatives for this purpose do exist, e.g., TypeCraft or Xigt, based on XML technologies. Another possibility would be to represent this type of data in RDF. Moreover, in order to make it truly interoperable, there should exist a standard vocabulary for this type of data. This paper introduces such a vocabulary, Ligt, an LLOD-native vocabulary for representing Interlinear Glossed Text as RDF data.

The paper is organized as follows. We describe existing approaches to representing IGT data in Section 2. We describe the Ligt core vocabulary in Section 3. Converters between Ligt and several popular formats are described in Section 4. Finally, Section 5 concludes the paper, outlining its main outcomes.

2 Data models for IGT data

2.1 Existing formats

This section describes several widely used formats for storing IGT data, including existing Linked Data representations.

There is a great variation in the ways to store IGT data. Probably, the biggest factor influencing the format is whether the data is stored for research or published alongside the research. In the latter case, it is usually a part of a book or a paper either scattered thorough the text or given as an appendix. This representation is not truly machine-readable, so this format cannot be considered reusable. However, there are initiatives in order to overcome this

[1] An example from the second author's fieldwork materials.

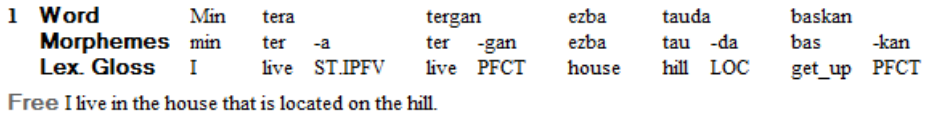


Figure 1 Mishar IGT sample, FLEx print view.

problem by making this data accessible, namely, ODIN, the Online Database of Interlinear Text [7],[2] which was created by parsing scholarly documents on the Web in order to extract such data. For storing the data, ODIN uses Xigt (eXtensible Interlinear Glossed Text).[3] Xigt is an XML-based data model created to simplify the format of IGT in most of the cases, allowing to scale up to accomodate different kinds of annotations. Figure 11 illustrates its XML data model.

IGT data used in research is mostly generated by field linguists working with (native) speakers during their work, so the format depends on what tools do they use. The most widely known applications developed specifically for creating IGT are Toolbox (formerly Shoebox)[4] and FLEx, its successor[5]. They both provide advanced functionality to enter and store IGTs, perform analyses, and build dictionaries. Both have their own advantages, and therefore, both are actively used in the community.

In our previous work, we introduced a shallow RDF representation for both FLEx and Toolbox formats [1]. Below, we briefly describe the respective data models, both in their original XML serialization and in a shallow RDF reconstruction. Other formats do exist, as well, e.g., TypeCraft, and different proposals within the Text Encoding Initiative (TEI).

2.2 RDF reconstruction of FLEx and Toolbox

The FLEx framework stores linguistic data as a set of XML documents: an XML file with all the texts with their markup and a number of auxiliary files: language settings, project settings, etc. The main file consists of a number of `<rt>` elements, each representing a database record. Hierarchy is established by linking records using the attribute `ownerguid` which references the parent record of the element. Records may consist of different elements depending on their `class` attribute.

Figure 1 shows selected glosses in the FLEx graphical user interface and Fig. 2 provides a fragment from its XML representation.

Another way of accessing FLEx data is to export its texts. Unlike the database-like structure of the main XML format, the format for exporting is hierarchical, and its semantics is more clear. FLEx distribution includes a (non-validating) XSD schema that illustrates the basic data structure of these files. Fig. 3 provides a fragment from the XML representation of the same fragment as in Fig. 2.

In [1], we propose a shallow RDF model based on the latter XML representation. Exporting texts from the main project XML leads to information loss which does not allow converting the result back to FLEx projects. Despite that, in this paper we employ this model as a basis for further conversion, while dealing with the main project format is currently under development.

[2] http://depts.washington.edu/uwcl/odin/
[3] https://github.com/xigt/xigt
[4] http://www-01.sil.org/computing/catalog/show_software.asp?id=79
[5] http://fieldworks.sil.org/flex

```
<rt class="StTxtPara"   guid="fa70b76b-5e19-48f6-aa14-9f8d4029ad95"
                        ownerguid="bada85a7-c2cd-47c2-9ad1-be46c3160e5f">
  <Contents>
    <Str>
      <Run ws="tat-Latn-RU-x-mishar">Min tera tergan ezba tauda baskan.</Run>
    </Str>
  </Contents>
  <ParseIsCurrent val="True"/>
  <Segments>
    <objsur guid="a5720126-e1f6-4b19-a5bb-74527f7c57f8" t="o" />
  </Segments>
</rt>
<rt class="Segment" guid="a5720126-e1f6-4b19-a5bb-74527f7c57f8"
                    ownerguid="fa70b76b-5e19-48f6-aa14-9f8d4029ad95">
  <Analyses>
    <objsur guid="248b923a-8d10-49a2-8979-db26d3125ead" t="r" />
    <objsur guid="c2483bb6-6a2d-4fa0-993c-56b538a92b42" t="r" />
    <objsur guid="821aa99b-c8f2-4be0-9634-24797addbb6b" t="r" />
    <objsur guid="c90f7e22-9397-49a7-b727-0cbec202392f" t="r" />
    <objsur guid="fd0fe7c2-b7c0-4b71-bad8-992e1100eb30" t="r" />
    <objsur guid="be7cdaea-8262-4367-a538-b7da11521aa1" t="r" />
    <objsur guid="0534cafe-d520-4548-bc47-962509b77f9f" t="r" />
  </Analyses>
...
```

Figure 2 Mishar IGT sample, FLEx XML.

The RDFS data model that we take as a basis for FLEx data conversion, and the aforementioned data fragment converted with respect to this data model are illustrated in Figs. 5 and 4, respectively.

In earlier work, we have shown that the FLEx conversion can also be applied to Toolbox data [1]. Together with FLEx, Toolbox is the most tool popular for working with IGT data is Toolbox. It is a predecessor of FLEx, although in some aspects it is more powerful than FLEx, so it is still widely used by field linguists. Most notably, it allows creating any number of user-defined "markers" (glossing/annotation layers) such as multiple orthographies or different variants of morphological glossing[6]. Given this, even though there is a process of importing Toolbox data into FLEx, it is not universally possible to do this without information loss.

Toolbox stores its data in an SFM[7] format. It is a text-based format where each line represents a layer defined by its marker at the beginning. Interlinear alignment is achieved by using the precise number of spaces: Each new segment on corresponding lines starts at the same position[8].

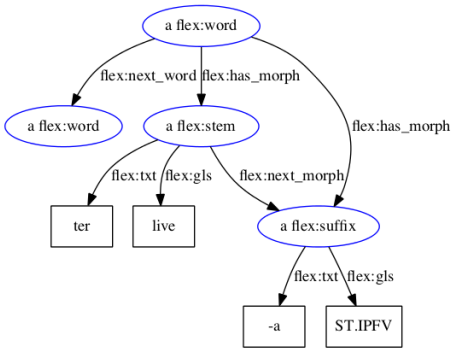
An existing shallow RDF representation for Toolbox resembles the one for FLEx with two key differences:

1. There is no paragraph division in Toolbox data hence `flex:has_phrase` relations can be directly between `flex:interlinear-glosses` and `flex:phrase`.
2. Triples with information regarding toolbox markers are stored in their own namespace since they can differ from the FLEx markers.

[6] The new version of FLEx, which is now available as beta, has similar functionality. However, the current stable version is still widely used.
[7] Standard Format Markers
[8] Due to historical pre-Unicode reasons, the position is calculated in the number of bytes, not in the number of characters.

```
<phrases>
  <phrase guid="a5720126-e1f6-4b19-a5bb-74527f7c57f8">
    <item type="segnum" lang="en">1</item>
    <words>
      ...
      <word guid="c2483bb6-6a2d-4fa0-993c-56b538a92b42">
        <item type="txt" lang="tat-Latn-RU-x-mishar">tera</item>
        <morphemes>
          <morph type="stem" guid="11e7849e-3aad-4c58-bdf4-966f783e6904">
            <item type="txt" lang="tat-Latn-RU-x-mishar">ter</item>
            <item type="gls" lang="en">live</item>
          </morph>
          <morph type="suffix" guid="11e7849e-3aad-4c58-bdf4-966f783e6904">
            <item type="txt" lang="tat-Latn-RU-x-mishar">-a</item>
            <item type="gls" lang="en">ST.IPFV</item>
          </morph>
        </morphemes>
      </word>
      ...
    </words>
    <item type="gls" lang="en">I live in the house that is located on the hill.</item>
  </phrase>
</phrases>
```

Figure 3 Mishar IGT sample, FLEx XML export.




```
:c2483bb6-6a2d-4fa0-993c-56b538a92b42
    a flex:word;
    flex:txt "tera"@tat-Latn-RU-x-mishar;
    flex:has_morph  :11e7849e-3aad-4c58-bdf4-966f783e6904_1,
                    :11e7849e-3aad-4c58-bdf4-966f783e6904_2;
    flex:next_word  :821aa99b-c8f2-4be0-9634-24797addbb6b.

:11e7849e-3aad-4c58-bdf4-966f783e6904_1
    a flex:morph, flex:stem;
    flex:txt "ter"@tat-Latn-RU-x-mishar;
    flex:gls "live"@en;
    flex:next_morph :11e7849e-3aad-4c58-bdf4-966f783e6904_2.

:11e7849e-3aad-4c58-bdf4-966f783e6904_2
    a flex:morph, flex:suffix;
    flex:txt "-a"@tat-Latn-RU-x-mishar;
    flex:gls "ST.IPFV"@en.
```

Figure 4 FLEx IGT sample, generated RDF graph.

2.3 RDF reconstruction of Xigt

In order to provide a generalization over existing, tool-specific data models, Ligt is being defined as a generalization over two data models, FLEx/Toolbox data structures and the data model of the Xigt format. With respect to Xigt, we use a shallow RDF reconstruction of the Xigt data model as a basis, akin to FLEx RDF for FLEx and Toolbox.

Xigt differs from tool-specific formats such as FLEx and Toolbox in that it aims to provide a generic data model for IGT data rather than to provide a serialization for an existing tool. The Xigt format was designed from scratch, it was explicitly intended to be easily extensible for different types of annotations, and thus differs greatly from FLEx or Toolbox formats. Xigt was designed as an XML format, and Fig. 6 illustrates the reconstruction of its structure as an RDFS schema:

- The top-level element of a Xigt document is a `xigt-corpus`, which contains `igt` elements that convey the actual annotation.
- An `igt` contains a number `tier` elements, each corresponding to a single layer of annotation. Each `tier` consists of several `items`.
- An `item` can contain text and carry additional attributes that contain the actual annotation, rendered here as datatype properties of the same name, e.g., `tag`.

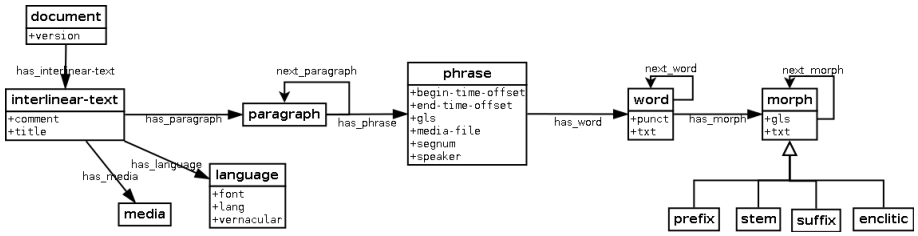


Figure 5 RDF schema fragment for FLEx data.

- Alignment between items is established by *alignment expressions* stored in items' attributes. Those expressions can refer either to one or more items or their parts: `p[1:3]` corresponds to characters 1–3 of the item `p` and `p1+p2[0:2]` corresponds to the full value of item `p1` and characters 0–2 of the item `p2`.
- The sequential order of `igt`, `tier` and `item` is inherent to the XML model, but must be explicated in the RDF rendering. For this purpose, we introduce the property `next`.
- For Xigt XML elements that contain a reference to (the id of) another Xigt XML element, we create an object property of the same name (e.g., `dep` for the annotation of dependency syntax).
- Any `xigt-corpus`, `igt` or `tier` can carry a `metadata` property with a `Metadata` object (corresponding to the `Metadata` element in Xigt/XML).
- The property `meta` assigns a `Metadata` object an XMLLiteral. Normally, this property is not to be used directly, but subproperties are to be created for different types of metadata. These subproperties of `meta` are derived from the `@type` attribute of `meta` elements in Xigt/XML.
- The Xigt RDFS vocabulary does not define subclasses of `igt`, `tier` and `item`, but such subclasses are expected to be defined by different applications, e.g., designated tiers for word segmentation, and morphological segmentation. In Xigt/XML, this is expressed with a `@type` attribute and we expect to derive such more specific subclasses from `@type`.
- In order to ground Xigt/RDF in existing web vocabularies, we define `tier` and `item` as `nif:String`s and postulate a `nif:subString` relation between them [5].
- Xigt elements are identified by a URI. If the Xigt XML element provides an `@id` attribute, this will be adopted as local name and combined with the document/graph URI. Otherwise, URIs are inferred from the structure of the Xigt XML file.

Along with converters for FLEx, Toolbox and other formats, we provide a converter from Xigt to Xigt/RDF as part of our LLODifier suite.[9]

For illustration, we provide a simple sample of the ODIN data base, v.2.3 (from by-doc-id/xigt/10.xml, see appendix for the original XML).

```
:igt10-6 a xigt:igt.
:igt10-6  xigt:metadata :meta1.
:igt10-6 xigt:has_tier :tier_18.

:tier_18 a xigt:odin_tier;
xigt:has_item :item_47, :item_48, :item_49.
```

[9] `https://github.com/acoli-repo/LLODifier`

```
:item_47 a xigt:item;
xigt:line "103";
xigt:tag "L";
xigt:odin_text "Ahmet hizli     ko-uyor-du";
xigt:next :item_48.

:item_48 a xigt:item;
xigt:line "104";
xigt:tag "G";
xigt:odin_text "Ahmet quickly run-PROG-PAST.3sg";
xigt:next :item_49.

:item_49 a xigt:item;
xigt:line "105";
xigt:tag "T";
xigt:odin_text "Ahmet was running quickly.".
```

In more complex examples (omitted here for reasons of space), `item`s are further split into morph(eme)s and the analysis is aligned across different tiers. An advantage of Xigt is that it allows to represent IGTs both in a fine-grained manner (as known from FLEx) and in such a coarse-grained way (as in the ODIN data, adopted here because of difficulties to infer morpheme-level alignment).

Aside from being more scalable with respect to its level of detail, Xigt differs greatly from the FLEx data model described above in the following aspects:

- It lacks any data structures that aggregate IGTs into larger groups such as paragraphs.
- It does not define formal data types for standard components of IGT analysis. Instead, these have to be defined by the data provider (via the `@type` attribute resp. `rdfs:subClassOf`).
- It does provide a complex mechanism for expressing and resolving alignment. In FLEx, this is restricted to substrings.
- It does not provide a vocabulary for metadata properties. Instead, these have to be defined by the data provider (via the `@type` attribute resp. `rdfs:subClassOf`).
- It does provide some properties that exceed the capability of traditional IGTs (e.g., `dep` for depedency syntax).

Like Toolbox (and – to a limited extent – FLEx), Xigt allows to add novel attributes/properties, it is, however, more generic, and allows to include other aspects of linguistic annotation. At the same time, it is underspecified with respect to its concepts: As the comparison with FLEx RDF shows, its data structures are also weaker, in that no vocabulary for essential categories in IGT annotation are provided, most notably words (tokens) and morph(eme)s. We design the Ligt vocabulary as a compromise between both extremes: A vocabulary that provides obligatory IGT data structures (as FLEx), but with the potential for further extensions and underspecification (as Xigt).

3 A native LLOD vocabulary for interlinear glossed text

We motivate Ligt as an abstraction over two application-specific data models, FLEx and Xigt, resp., the RDF vocabularies created for expressing their information in RDF. We see the main contribution of our paper in the formulation of this vocabulary, as a basis for an exchange and publication format for interlinear glossed text in the web of data, and for a

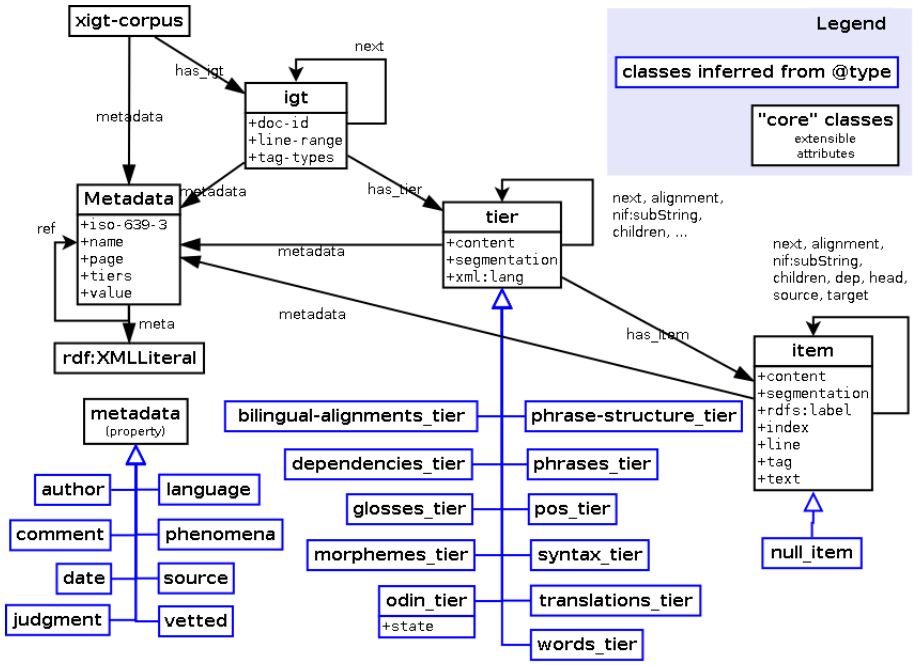


Figure 6 Xigt RDFS data model, also cf. Fig. 11, p. 15 for the underlying XML Schema.

tool chain developed for such data, e.g., converters from classical IGT formats via Ligt to other annotation formats, e.g., tabular formats such as used by ELAN. As part of such a tool chain, we provide a converter suite that generates Ligt from FLEx, Toolbox and Xigt data, and by means of SPARQL, a generic functionality to generate TSV exports from Ligt is already provided by off-the-shelf technology.

In addition, we see an important contribution of this vocabulary as an input to the development of specifications for morpheme-level analyses for W3C vocabularies for lexical data[10] and linguistic annotations.[11]

3.1 Core vocabulary

Ligt vocabulary defines classes and properties to describe the relations between the documents, morphemes and their annotations in texts with an interlinear glossing. This vocabulary is a generalization over two shallow RDF representations introduced in the previous section: a model for data from FLEx or Toolbox, and Xigt RDF model. Other than these, however, it is defined independently from an existing tool chain.

In order to develop an interoperable solution, the base classes are derived from two widely used external vocabularies: Dublin Core [12][12] and the NLP Interchange Format [6, NIF].[13]

[10] Note the development of a morphology module within the OntoLex vocabulary `https://www.w3.org/community/ontolex/wiki/Morphology`.

[11] At the time of writing, morphology is not adequately covered by existing RDF vocabularies for linguistic annotations: NIF [6] and NIF-based vocabularies such as ITS [4] focus on annotations at the level of words or larger, Web Annotation [11] does not provide a designated vocabulary for linguistic annotation at all.

[12] `http://purl.org/dc/terms/`

[13] `http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core/nif-core.html`

`ligt:Document`. A document is a subclass of `dc:Dataset` that represents a collection of interlinear glossed texts. We formulate no constraints on the nature of documents, a document may be the electronic edition of a set of coherent texts, but also an unstructured collection of isolated examples. A `ligt:Document` is equivalent to a `flex:document`, and closely resembles `xigt:corpus`.

`ligt:has_text`. A document must have at least one `ligt:has_text` property that points to a `ligt:Segment`, e.g., an object of type `ligt:InterlinearText`. The property is closely related with `flex:has_interlinear_text`.

`ligt:Segment`. A segment is a `nif:String` that is an abstraction over (interlinear) text, paragraph and utterance. Segments that contain each other are connected by `nif:subString`.

- `ligt:InterlinearText`. An interlinear text is a coherent sequence of interlinear glosses, and defined as a `dc:Text` and equivalent with `flex:interlinear-text`. There is no exact pendant of `ligt:InterlinearText` in Xigt. `ligt:InterlinearText` is equivalent to a `ligt:Segment` without another `ligt:Segment` that it is a `nif:subString` of.
- `ligt:Paragraph`. A paragraph is a `nif:Paragraph` within a `ligt:InterliearText` that groups together multiple utterances or other segments. It corresponds to `flex:paragraph`, no pendant in Xigt. Paragraph is equivalent with a `ligt:Segment` that is neither `ligt:InterlinearText` nor `ligt:Utterance`.
- `ligt:Utterance`. An utterance is a coherent, consecutive sequence of words, as typically produced by a single speaker in a communication situation. The notion of utterance is closely related to a `nif:Sentence`, but we do not require utterances to be sentential in a syntactic sense. We define an utterance as a `ligt:Segment` without further `ligt:Segments` as `nif:subString`. Utterance is equivalent to `flex:phrase`. There is no exact pendant of `xigt:igt` in Ligt, but every `xigt:igt` is both a `ligt:Utterance` and a `ligt:InterlinearText`.

`ligt:hasTier`. This property assigns an utterance a tier that contains its annotations. Corresponds to `flex:has_tier`.

`ligt:Tier`. A tier is a set of annotations that share the same characteristics, and in particular, the same segmentation. Tier corresponds to `xigt:tier`, there is no exact equivalent in FLEx, as FLEx considers tier definitions as being inherent in the notions of `flex:phrase`, `flex:word` and `flex:morph`. Based on FLEx data structures, two subclasses of tier are provided:

- `ligt:WordTier`. A tier adopting a segmentation into words (i.e., `flex:words`).
- `ligt:MorphTier`. A tier adopting a segmentation into morph(eme)s (i.e., `flex:morphs`).

Note that unlike FLEx, Ligt does not posit a uniqueness constraint on these tiers, but instead supports, for example, to have multiple tiers for morphs at different granularity. Ligt also permits to provide application-specific tiers (as currently by Xigt XML `@type` attributes).

We define a tier as a `nif:String` and its items as the corresponding `nif:subStrings`.

`ligt:item`. Property assigning a `ligt:Tier` a `ligt:Item`. Corresponds to `xigt:has_item`. As both `ligt:Tier` and `ligt:Item` are defined as `nif:Strings`, this property is defined as a subproperty of `nif:subString`.

`ligt:Item`. Abstract class representing elements of a `ligt:Tier`, representing the unit of annotation in an IGT. Equivalent to `xigt:item`, and likewise defined as a subclass of `nif:String`. It is possible to provide application-specific subclasses (as by `@type` in Xigt).

Two following pre-defined subclasses are provided:

`ligt:Word`. A grammatical or orthographic word as the basis for further annotation, equivalent with `nif:Word`. A `ligt:WordTier` is defined as a `ligt:Tier` for which every `item` is a `ligt:Word`. Roughly equivalent with `flex:word`, but note that Ligt (unlike FLEx) does not prohibit concurrent word segmentations of the same utterance.

`ligt:Morph`. We define a morph as a `nif:String` that corresponds to the smallest unit of grammatical analysis applicable to a given word. A `ligt:MorphTier` is defined as a `ligt:Tier` for which every `item` is a `ligt:Morph`. Roughly equivalent with `flex:morph`, but note that Ligt (unlike FLEx) does not prohibit concurrent word segmentations of the same utterance.

An item can be a `nif:subString` of another item at another tier; this is the preferred way to express that a `ligt:Morph` is contained in a `ligt:Word` (cf. `flex:has_morph`).

`ligt:next`. Presents the sequential order of items, corresponds to `xigt:next` and `flex:next_word` and `flex:next_morph`.

Ligt is grounded in the generalization over (the RDF vocabularies inferred for) FLEx and Xigt, but the concept of tiers is exclusive to Xigt, so there is no straightforward way to generalize this concept for both representations. In order to do this, we introduced a base class `ligt:Tier` and two subclasses: `ligt:WordTier` and `ligt:MorphTier` which should correspond to sequences of words and morphs, respectively. Tiers in Ligt must consist of elements on the same level of granularity hence we merge Xigt tiers with identical segmentation. Xigt has been designed for reversible IGT parsing. This means that it provides a standoff mechanism that refer to segments and annotation values rather than providing them. In Xigt RDF, these are resolved, but xigt:content and xigt:alignment are preserved. In the generalization, these are no longer necessary. They should not be deleted, though, as they cannot be easily reproduced. But they provide Xigt-specific information and do not need to be represented in the overarching model.

Both `ligt:Word` and `ligt:Morph` are subclasses of `ligt:Item` and are objects of a property `ligt:item` for the word and morph tiers, respectively. Finally, for compatibility with FLEx, we introduce subclasses of `ligt:Morph` for representing prefixes, suffixes, stems and enclitics.

The data model for text representation in Ligt is illustrated in Fig. 7.

3.2 Metadata

Every concept identified above can be subject to metadata annotations. In order to provide a consistent domain definition for such properties, we introduce `ligt:Element` as an abstract superclass over `ligt:Document`, `ligt:Segment`, `ligt:Tier` and `ligt:Item`.

FLEx metadata is represented by (a fixed set of) simple properties (`flex:version`, `flex:comment`, `flex:title`, `flex:has_media`, `flex:has_language`, etc.) for which no generalization is provided. In opposition to that, Xigt metadata is modelled by means of reification, with a `xigt:Metadata` object mediating between metadata properties and its target.[14] In Ligt, we thus support both mechanisms, but we do not prescribe any specific metadata vocabulary. Instead, any metadata property must be a subproperty of

[14] In Xigt XML, metadata is represented by the container element `metadata` that groups together several `meta` statements. As the `metadata` element can carry its own XML attributes, it has to be rendered as reification in Xigt RDF.

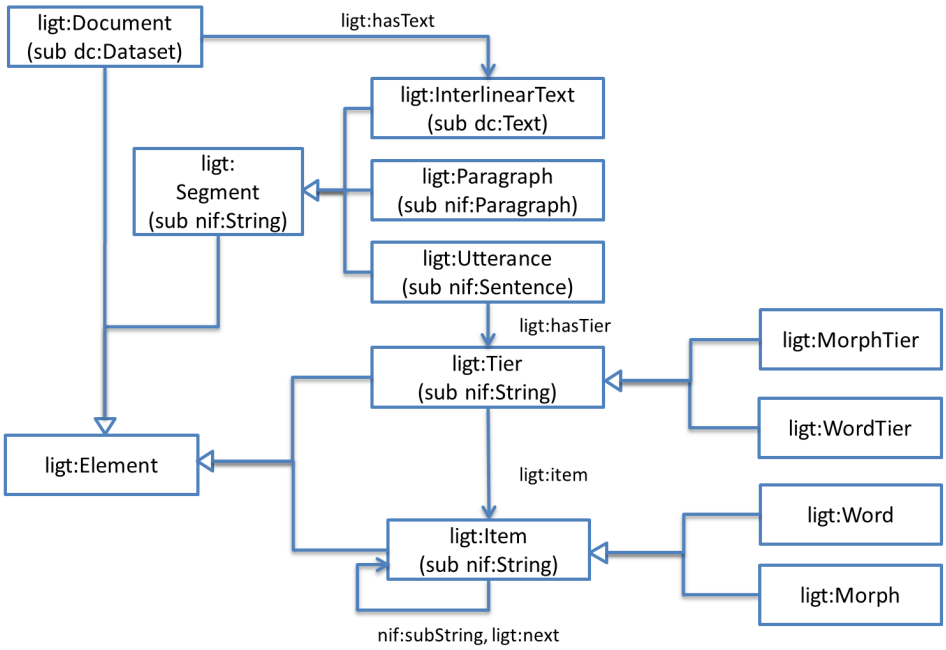


Figure 7 Ligt data model, excluding metadata.

`ligt:metadata`, and any metadata object must be an instance of `ligt:Metadata`. As for the reified representation of metadata, we follow the Web Annotation data model [11].

`ligt:metadata`. Abstract datatype property that assigns a `ligt:Element` a literal value. Superproperty of `flex:version`, etc.
`ligt:Metadata`. Subclass of `oa:Annotation` that represents the reification of `ligt:metadata`.
`ligt:metaTarget`. Subproperty of `oa:hasTarget` that points from a metadata object to the `ligt:Element` that the metadata refers to. Corresponds to the inverse of `xigt:metadata`.
`ligt:metaBody`. Subproperty of `oa:hasBody` that connects a `ligt:Metadata` object with the literal value that contains the metadata. Superproperty of `xigt:iso-693-3`, etc.
`ligt:refBody`. Subproperty of `oa:hasBody` that connects a `ligt:Metadata` object with another (metadata) object. Corresponds to `xigt:ref`.

The Xigt data model allows complex metadata attached to any `corpus`, `igt` or `tier` element. It can be both simple values like language, source or date and complex structures. In the shallow RDF representation this was modeled with reification, where multiple attributes for the same type of metadata are represented as a collection. This approach is powerful, but does not make much sense for atomic metadata properties from FLEx data model. In order not to overcomplicate the model, we decided to use both RDF reification to express the complex Xigt metadata and the more transparently structured FLex metadata. This will keep the model simple but retain its expressivity.

In order to be able to link metadata to elements on different level, we define a top-level concept `ligt:Element`, which is the domain of `ligt:metadata` property. Top-level class `ligt:Document` is defined as its subclass. Atomic metadata elements should be subclasses of `ligt:annotation` whereas complex metadata should be a subclass of `ligt:metadata`. The reified representation is modeled with properties `ligt:metaTarget` and `ligt:metabody`, which are derived from `hasTarget` and `hasBody` properties of the OpenAnnotation vocabulary [10].

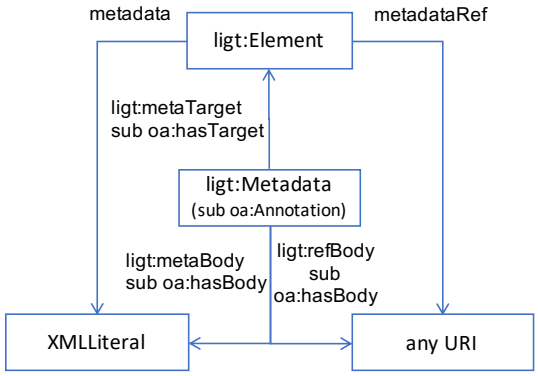


Figure 8 Ligt metadata representation.

The metadata part of the model is illustrated in Fig. 8.

4 Implementation

Here we describe the problems with converters we developed between the shallow RDF representations and Ligt. Mainly, the conversion is performed by a series of SPARQL updates. In the following paragraphs we briefly discuss our decisions and difficulties with the conversion.

Xigt → Ligt. When converting to Ligt, we drop explicit information about segmentation and alignment, since this is already resolved, and we want to keep the data as general as possible. For the same reason we drop tier ordering information (property `xigt:nextTier`), since the ordering of tiers is specific only to Xigt. We also need to convert the metadata information to the OpenAnnotation metadata model.

Xigt ← Ligt. The main problem with the conversion in this direction is that Ligt omits tier ordering information even if the data was originally got converted from Xigt. In order to convert, the order of tiers should be specified manually or left in the default order and then get reordered later by other means.

If the data originally came from FLEx or Toolbox, there may be information about paragraphs or texts, which cannot be expressed in terms of Xigt, which means that this information will be ignored.

FLEx ↔ Ligt. In this conversion, the main difficulty is transforming the data properties into metadata, for instance, language information stored in FLEx data model should become metadata in Ligt representation and vice versa. Another thing is the introduction of fictitious elements within multiple conversions, e.g. putting the text in the paragraph even if there was no paragraphs in the original data when converting to FLEx RDF.

One application of Ligt is to facilitate the integration and the querying of IGT from various sources. The concept is illustrated in Fig. 9. At the moment, we provide converters from FLEx, Toolbox and Xigt to FLEx RDF and Xigt RDF as part of our LLODifier repository,[15] respectively, as well as SPARQL Update scripts to convert that data into Ligt.[16]

[15] `https://github.com/acoli-repo/LLODifier`
[16] `https://github.com/acoli-repo/ligt`

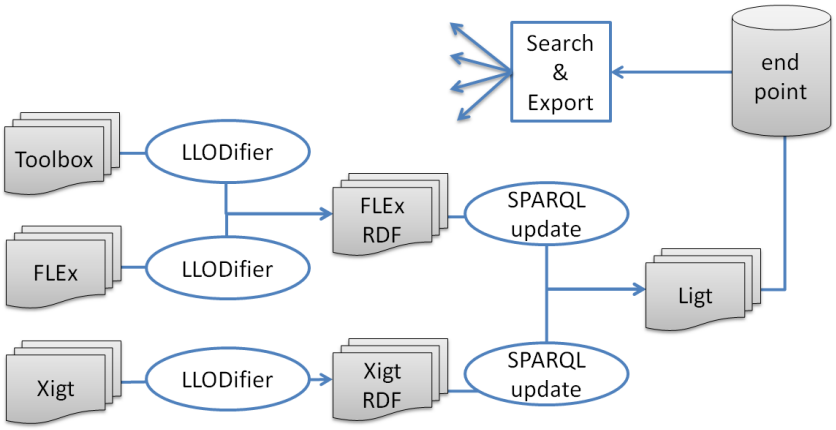


Figure 9 Ligt-based IGT processing workflow.

We have converted the ODIN v.2.3 into Ligt, with a total of 7.5 million triples for 158 007 IGTs for 2 888 language varieties. Interfaces tailored towards the needs of end users (linguists) are still under development.

5 Summary and outlook

In this paper we presented Ligt, the first LLOD-native IGT vocabulary for LLOD-data, based on three formats. This vocabulary is grounded in widely used vocabularies (Dublin Core, NIF and WebAnnotation), but extends them with respect to the coverage of morphology. With Ligt, we aim to achieve the following goals:

Provide a vocabulary for publishing and sharing IGT data via the (Linguistic) Linked Open Data cloud.

Contribute to the extension of W3C vocabularies such as Ontolex-lemon with respect to the coverage of morphology.

Trigger the development of morphology-aware vocabularies for the representation of corpora and linguistic annotations.

Prepare the ground for developing an infrastructure for the integrated querying and processing of IGTs and related linguistic data.

Publishing interlinear glosses as LLOD facilitates their reusability and interoperability, allowing querying several IGT datasets at once, linking them to external resources and more. At the same time, using different shallow data models, each of which inherits conceptual model of the corresponding framework, is not enough to achieve true interoperability. All three frameworks provide slightly different set of functions, and the conceptual model behind their data representation differ greatly. Even though this shallow approach guarantees data structures that are transparent and familiar to their user community, it does not provide the rich semantics of more advanced vocabularies for language resources.

By creating a universal vocabulary for modeling IGT annotations, and creating converters from those three formats to this unified representation should improve interoperability further.

Given this, the main contribution of this paper is a proposal of an RDF-native data model that not only allows to unify IGT data developed under different frameworks to a completely new level, but also allows to generalize to other use cases in linguistics, as well. Beyond

establishing structural (format) interoperability by means of a common data representation, our approach also allows to make use of shared vocabularies and terminology repositories available from the (Linguistic) Linked Open Data cloud, e.g., for representing language varieties [9], linguistic phenomena [2], or lexical information [8].

References

1 Christian Chiarcos, Maxim Ionov, Monika Rind-Pawlowski, Christian Fäth, Jesse Wichers Schreur, and Irina Nevskaya. LLODifying linguistic glosses. In *Proceedings of Language, Data and Knowledge (LDK-2017)*, Galway, Ireland, June 2017.

2 Christian Chiarcos and Maria Sukhareva. OLiA - Ontologies of Linguistic Annotation. *Semantic Web Journal*, 518:379–386, 2015.

3 Bernard Comrie, Martin Haspelmath, and Balthasar Bickel. The Leipzig Glossing Rules: Conventions for interlinear morpheme-by-morpheme glosses. `https://www.eva.mpg.de/lingua/pdf/Glossing-Rules.pdf`, 2008.

4 David Filip, Shaun McCance, Dave Lewis, Christian Lieske, Arle Lommel, Jirka Kosek, and Felix Sasaki. Internationalization Tag Set (ITS) Version 2.0. Technical report, W3C Recommendation, 2013.

5 Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating NLP using Linked Data. In *12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia*, 2013. URL: `http://svn.aksw.org/papers/2013/ISWC_NIF/public.pdf`.

6 Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating NLP using Linked Data. In *Proc. 12th International Semantic Web Conference, 21-25 October 2013*, Sydney, Australia, 2013. also see `http://persistence.uni-leipzig.org/nlp2rdf/`.

7 William D. Lewis. ODIN: A model for adapting and enriching legacy infrastructure. In *Second International Conference on e-Science and Grid Technologies (e-Science 2006), 4-6 December 2006, Amsterdam, The Netherlands*, page 137. IEEE Computer Society, 2006. `doi:10.1109/E-SCIENCE.2006.106`.

8 John P. McCrae, Julia Bosque-Gil, Jorge Gracia, Paul Buitelaar, and Philipp Cimiano. The Ontolex-Lemon model: development and applications. In *Proceedings of eLex 2017 conference*, pages 19–21, 2017.

9 Sebastian Nordhoff and Harald Hammarström. Glottolog/Langdoc: Defining dialects, languages, and language families as collections of resources. In *First International Workshop on Linked Science 2011-In conjunction with the International Semantic Web Conference (ISWC 2011)*, 2011.

10 Robert Sanderson, Paolo Ciccarese, and Herbert Van de Sompel. Open Annotation Data Model. Technical report, W3C Community Draft, 08 February 2013, 2013.

11 Robert Sanderson, Paolo Ciccarese, and Benjamin Young. Web Annotation Data Model. Technical report, W3C Recommendation, 2017.

12 Stuart Weibel, John Kunze, Carl Lagoze, and Misha Wolf. RFC 2413 - Dublin Core metadata for resource discovery. URL `http://www.ietf.org/rfc/rfc2413.txt` (July 31, 2012), September 1998. Network Working Group.

A Xigt XML data

In the paper, we presented Xigt RDF as one starting point for the development of Ligt. In order to relate this to actually available Xigt data, the interested reader may be interested in the original XML schema and sample data, provided here in Figs. 11 and 10, respectively. Additional documentation can be found under `https://github.com/xigt/xigt/wiki/DataModel`.

```
<igt id="igt10-6" doc-id="10" line-range="103-105" tag-types="L G T">
   <metadata>
     <meta id="meta1">
       <dc:subject olac:code="tur" xsi:type="olac:language">Turkish</dc:subject>
       <dc:language olac:code="en" xsi:type="olac:language">English</dc:language>
     </meta>
   </metadata>
   <tier id="n" type="odin" alignment="c" state="normalized">
     <item id="n1" alignment="c1" line="103" tag="L">Ahmet hizli     ko-uyor-du</item>
     <item id="n2" alignment="c2" line="104" tag="G">Ahmet quickly run-PROG-PAST.3sg</item>
     <item id="n3" alignment="c3" line="105" tag="T">Ahmet was running quickly.</item>
   </tier>
</igt>
```

Figure 10 Xigt XML sample data, ODIN v. 2.3, file by-doc-id/xigt/10.xml.

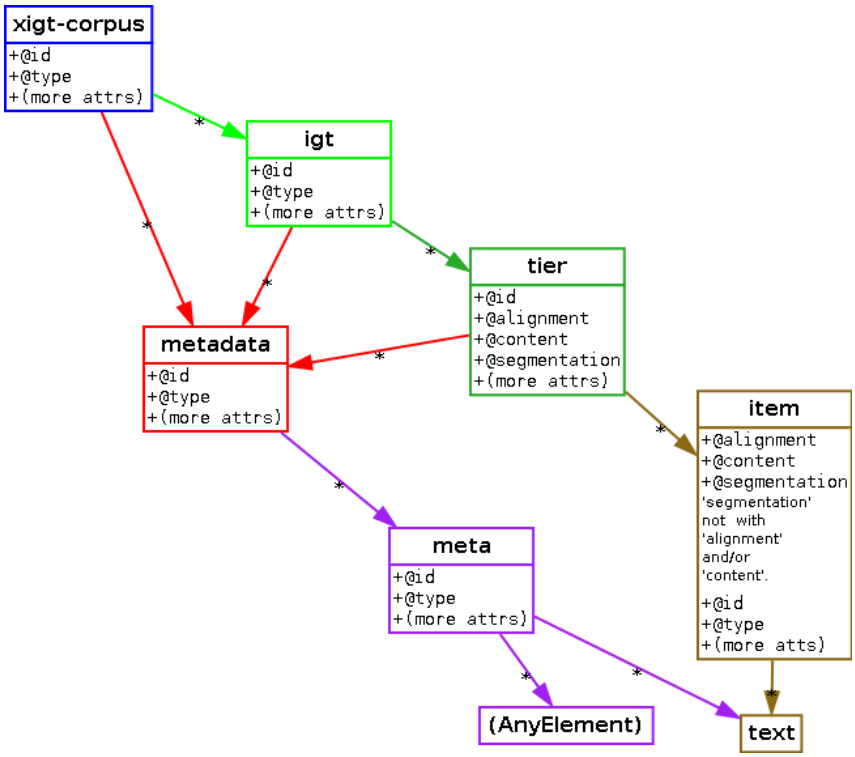


Figure 11 Xigt XML Schema.

The Shortcomings of Language Tags for Linked Data When Modeling Lesser-Known Languages

Frances Gillis-Webber
Library and Information Studies Centre, University of Cape Town, South Africa
http://www.dkis.uct.ac.za
fran@fynbosch.com

Sabine Tittel
Heidelberg Academy of Sciences and Humanities, Germany
http://www.deaf-page.de
sabine.tittel@urz.uni-heidelberg.de

Abstract

In recent years, the modeling of data from linguistic resources with Resource Description Framework (RDF), following the Linked Data paradigm and using the OntoLex-Lemon vocabulary, has become a prevalent method to create datasets for a multilingual web of data. An important aspect of data modeling is the use of language tags to mark lexicons, lexemes, word senses, etc. of a linguistic dataset. However, attempts to model data from lesser-known languages show significant shortcomings with the authoritative list of language codes by ISO 639: for many lesser-known languages spoken by minorities and also for historical stages of languages, language codes, the basis of language tags, are simply not available. This paper discusses these shortcomings based on the examples of three such languages, i.e., two varieties of click languages of Southern Africa together with Old French, and suggests solutions for the issues identified.

2012 ACM Subject Classification Computing methodologies → Language resources; Information systems → Dictionaries; Information systems → Semantic web description languages; Information systems → Graph-based database models; Information systems → Resource Description Framework (RDF); Software and its engineering → Interoperability; Information systems → Multilingual and cross-lingual retrieval; Computing methodologies → Information extraction; Computing methodologies → Artificial intelligence

Keywords and phrases language codes, language tags, Resource Description Framework, Linked Data, Linguistic Linked Data, Khoisan languages, click languages, N|uu, ǁ'Au, Old French

Digital Object Identifier 10.4230/OASIcs.LDK.2019.4

Acknowledgements We would like to thank the reviewers for helpful comments and insightful feedback.

1 Introduction

The publication of language data on the Web as Resource Description Framework (RDF), and according to Tim Berners-Lee's Linked Data principles[1], has contributed to the emergence of a multilingual web of data. Publishing language resources as Linked Data allows for language resources to be exploited with the benefits of structural interoperability (same format and query language leading to cross-resource access), conceptual interoperability (shared standard vocabularies), accessibility (via standard Web protocols), and resource integration (via linked resources) [6].

[1] https://www.w3.org/DesignIssues/LinkedData.html [10-01-2019].


© Frances Gillis-Webber and Sabine Tittel;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 4; pp. 4:1–4:15

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

After a brief introduction to RDF and Linked Data, particularly in the context of linguistic resources, as well as language codes and language tags (Section 1), we present the challenge addressed in this paper: finding solutions for the shortcomings of language tags when identifying near-extinct and historical languages (Section 2), and we do so by modeling data from three languages, e.g., two click varieties from the language family previously referred to as 'Khoisan', and Old French (Section 3). The paper concludes with a discussion of the findings (Section 4) and directions for future work (Section 5).

1.1 RDF and (Linguistic) Linked Data

RDF is the standard data model for resources of the Semantic Web [9]. It expresses data as *subject-predicate-object* triples to facilitate data interchange on the web. Each *subject* and *object* is a node; the *predicate* forms a relation (edge) between two nodes. The *subject* can be a URI (Uniform Resource Identifier) or a blank node, the *predicate* can only be a URI, and the *object* can be a URI, blank node or a literal (described as a string), see [9, 3].

Linked Data (LD) can be defined as the «set of best practices for publishing and connecting structured data on the Web», and it builds on the RDF data model using HTTP (Hypertext Transfer Protocol) URIs [35, 4-12]. The LD principles have been adapted in many fields, including linguistics, where it has led to the creation of numerous datasets published as Linguistic Linked Open Data (LLOD)[2]: lexicons, annotated corpora, dictionaries, etcetera ([4, 24]). The model that has become the *de facto* standard for describing linguistic resources is the OntoLex-Lemon vocabulary[3] [26, 587]. The focus within this field lies on well-resourced languages and, in particular, on their modern stages, with a small number of examples of linguistic resources documenting low-resourced languages (e.g., [13, 27, 15]) and also historical language stages (e.g., [32, 7, 22, 31, 2]).

1.2 Language codes → language tags

To use unique codes for the identification of languages is necessary for any environment that follows BCP 47 [28]. Examples include language identification in RDF and XML documents (the latter using the `xml:lang` attribute), and institutions such as language repositories, e.g., the Open Language Archives Community (OLAC) and the World Atlas of Language Structures (WALS).[4] A unique language code is able to disambiguate the case when one language name refers to several languages, and one language has several names.

A language code «represents one or more language names, all of which designate the same specific language» [19]. The International Organization for Standardization (ISO) provides a standard for language codes: ISO 639 with Parts 1–3. In principle, the language codes in each part «are open lists that can be extended and refined», and a Registration Authority nominated by ISO maintains each part [12]. ISO 639-1 provides a two-letter code and it is a subset of ISO 639-2, which provides a three-letter code allowing for more languages to be represented. Both ISO 639-1 and ISO 639-2 represent major languages that are most frequently expressed in the world's literature ([12, 18]). The individual languages in ISO 639-2 are in turn a subset of those in ISO 639-3 that aims «to give as complete a listing of languages as possible» [12]. The types of languages covered include living, extinct, ancient, historic and constructed languages; their scope can either be an individual language or a macrolanguage, and the modality is spoken, written or signed ([12, 18, 19, 20]).

---

[2] `http://linguistic-lod.org/` [26-12-2019].
[3] `https://www.w3.org/2016/05/ontolex/` [31-12-2018].
[4] `http://www.language-archives.org/`; `https://wals.info/` [15-03-2019].

For individual languages, only varieties which are considered to be distinct languages are represented in ISO 639-3, with any dialects encompassed within the language code of that language. The language code «represents the complete range of all the spoken or written varieties of that language, including any standardized form» [19]. A macrolanguage code represents a cluster of language varieties. Macrolanguages differ from language collections in that for the former, the languages must be deemed very closely related, and for the latter, there can be a loose relation, but there should be some connecting feature, be it historical, geographical, or a linguistic association [28, 33]; language collections are only represented in ISO 639-2, and macrolanguages are only represented in ISO 639-3 [19].

A language tag is similar in concept to a language code, except the latter can be used in any discipline, and the former is intended for the internet community. The scope of a language tag is defined by IETF's BCP 47. BCP 47 is a document which specifies Best Current Practice for tags for identifying languages, and the language in question is able to be refined further from the ISO 639 language code ([12]; [28, 1-4]; [21]). Language tags are of the form: *language-extlang-script-region-variant-extension-privateuse*, comprised of one or more sub-tags, each separated by a hyphen; *language* is the shortest language code from ISO 639, and the remaining sub-tags are distinguished from each other «by length, position in the tag, and content» ([21]; [28, 4]).

1.3 Language codes for linguistic resources

The OntoLex-Lemon specification requires each linguistic resource, be it a lexicon, a lexical entry, or a lexical concept, to be identified using a URI to the relevant ISO 639 code, with RDF requiring each string literal in an *object* to be 'language-tagged'.[5]

A language code (or tag) is thus used in the following scenarios:

1. to identify a lexicon:
   - when a triple with the predicate `dct:language`[6] is declared: this is to the URI of an ISO 639 language code [8];
2. to identify a lexical entry:
   - same as (1);
3. for the language tagging of string literals:
   - this is a language tag, which, in the absence of additional sub-tags, is an ISO 639 language code [9].

A lexical entry in RDF, described using OntoLex-Lemon and serialized in Turtle[7], can be modeled as follows:

```
@PREFIX ontolex: <http://www.w3.org/ns/lemon/ontolex#> .
@PREFIX lexinfo: <http://www.lexinfo.net/ontology/2.0/lexinfo#> .
@PREFIX dct:     <http://purl.org/dc/terms/> .
@PREFIX rdfs:    <http://www.w3.org/2001/02/rdf-schema#> .

:entry/en-n-bile a ontolex:LexicalEntry , ontolex:Word ;
   lexinfo:partOfSpeech   lexinfo:Noun ;
   dct:language           <http://id.loc.gov/vocabulary/iso639-2/en> ,
                          <http://lexvo.org/id/iso639-1/en> ;
   rdfs:label             "bile"@en ;
```

[5] `https://www.w3.org/2016/05/ontolex/#conventions-in-this-document` [10-01-2019].

[6] Beyond RDF, OntoLex-Lemon, and DublinCore (dct) vocabulary, we use classes and properties of LexInfo, RDFS, SKOS, and DBpedia, see the respective URLs within the code examples.

[7] Terse RDF Triple Language, an easy to read serialization of RDF statements, `http://www.w3.org/TR/turtle/` [11-01-2019].

```
11    ontolex:canonicalForm  :entry/en-n-bile#lemma ;
12    ontolex:sense          :entry/en-n-bile#sense1 ;
13    ontolex:evokes         :concept/000000001 .
```

Where:

- Point 2 is demonstrated in Line 8-9: the applicable language codes for the lexical entry, from ISO 639-2 and ISO 639-1 respectively, are indicated as 'English'.
- Point 3 is demonstrated in Line 10: the language of the literal "bile" is specified with the ISO 639-1 code for English.

2 The shortcomings of language tags

The ISO 639 standard list includes more than 6,900 language codes[8] but it neither covers all the world's languages nor all historical language stages of the languages. This is problematic when modeling under-resourced or extinct languages for which a language code does not exist. To the best of our knowledge, this problem has not been properly addressed in the literature. A recent email thread in the W3C Semantic Web forum[9] expressed the opinion to do away with language tags altogether, but there was not shared consensus on this point.

Chiarcos and Sukhareva [7] show the conversion of legacy data from dictionaries of the historical language stages of Germanic languages (Old Saxon, Old High German, Old Norse, etc.) and find the following compensation for the lack of language codes within ISO 639: they preserve the original language abbreviations of the dictionary resource and extract «all language identifiers, and by a hand-crafted mapping from the original abbreviations», ISO 639-3 codes are assigned where possible [7, 44b]. The language URIs are represented using *lexvo* [10], but «[u]nfortunately, many abbreviations could not be resolved against lexvo, in particular, this included hypothetical forms for reconstructed historical language stages, e.g., Proto-Germanic.» They conclude that the extension of existing terminologies with respect to historical language stages is a great desideratum [7, 44b]. Their approach results in code such as `lemon:language "ae."@deu`, with 'ae.' being the German abbreviation for *Altenglisch* in the dictionary resource [23], see [7, 44b], and 'deu' being the ISO 639-3 language code for Standard German.[10]

The same approach has been taken by Declerck et al. for the transformation of the data from the *Wörterbuch der bairischen Mundarten in Österreich* (WBÖ)[11] into LD [11]: in the code sample given at [11, 347], the language tag for the Bavarian language is modeled as a literal: `bar"^^xsd:string"`, which raises the question why it is not given in the form of `@bar`, 'bar' being the ISO 639 code for Bavarian.[12] One might speculate that this could serve as a means to distinguish the language documented in the WBÖ (Bavarian varieties spoken in Austria) from Bavarian spoken in Bavaria; however, the problem is not addressed in the paper.

Amongst the findings of Tittel and Chiarcos [32] is the fact that due to the lack of appropriate language codes, the problem of modeling the different dialectal forms of lexemes in linguistic resources of Old French is still unsolved: for the conversion of the data of

[8] According to the table in `https://de.wikipedia.org/wiki/ISO_639` [10-01-2018].
[9] Language-tagged strings Re: Towards easier RDF: a proposal [Electronic mailing list, 23-26 November] 2018, `https://lists.w3.org/Archives/Public/semantic-web/2018Nov/thread.html#msg90` [01-01-2019].
[10] `https://iso639-3.sil.org/code/deu` [11-01-2019] (639-1: 'de'; 639-2/B: 'ger').
[11] `https://wboe.oeaw.ac.at/` [11-01-2019].
[12] `https://iso639-3.sil.org/code/bar` [11-01-2019].

the *Dictionnaire étymologique de l'ancien français* (DEAF, [1]) following the Linked Data paradigm, the researchers established that all graphical variants of a given Old French lexeme could only be identified by ISO 639-3 code 'fro'[13] for overall Old French. This meant that information originally included in the linguistic resource such as 'Anglo-Norman' or 'medieval Lorraine' scripta[14] – information that is very valuable for the research of Old French dialects – would be excluded from the language description when converted to Linguistic Linked Data. To solve this problem, [32, 65] propose to define the code 'fro' in ISO 639-3 as a macrolanguage and to register the Old French dialects as varieties associated to 'fro'. (There had been an attempt to include varieties of historic languages within ISO 639-6, but this Part was withdrawn in 2014.[15])

Bellandi et al. [2] discuss the modeling of linguistic data from Old Occitan (a Romance language spoken during the Middle Ages in what is today southern France) and other languages using OntoLex-Lemon. To code their Old Occitan lexemes, they use the tag 'aoc': `lemon:writtenRep "canabo"@aoc` [2, 4]. One rightly assumes that this is the ISO 639-3 code 'aoc', however, 'aoc' represents the Pemon language of the Cariban language family, a language in Venezuela.[16] The correct ISO 639 code for the language is 'pro' (= Old Provençal, the former term for the language)[17], and presumably 'aoc' simply is an abbreviation for French *ancien occitan*. Their handling of the use of codes is illegal: the definition of a language tag using the '@' sign and a language code must be BCP 47 compliant to be valid.[18] [2] do not address this issue, nor do they address the issue of creating their own language codes.

We conclude that new language codes need to be created, in a way that adheres to current standards and best practices of language identification. The objective of this paper is to contribute to the discussion of this problem. On the basis of three example languages, we will propose solutions to meet the requirements of the languages discussed.

The following languages serve as our examples:

1. N|uu and ǁ'Au: two dialects from N‖ng, a critically endangered non-Bantu click language in Southern Africa, that are both near-extinct [30, 7].
2. Old French: the ancestor of modern French, spoken during the Middle Ages.

In our sample code, we will focus on language-tagged string literals. It is clear, however, that the described problems and proposed solutions also apply to language URIs for lexicons and lexical entries.

3 Finding solutions for N|uu and ǁ'Au, and Old French

We focus on varieties of N‖ng and Old French to underline the fact that they are good examples of the need to preserve the languages and their historical stages as a key to understanding our cultural heritage: language is the storehouse of our culture, both past and present. It captures all aspects of life. It is subject to change and, thus, mirrors the development of our culture, of our state of mind, and of our social interaction through time.

[13] `https://iso639-3.sil.org/code/fro` [07-01-2019].
[14] Scripta is the term for the written form of a spoken dialect. Anglo-Norman is one of the varieties of Old French; it was spoken in England during the Anglo-Norman period.
[15] `https://www.iso.org/standard/43380.html` [07-01-2019].
[16] `https://iso639-3.sil.org/code/aoc`, `https://www.ethnologue.com/language/aoc` [11-01-2019].
[17] `https://iso639-3.sil.org/code/pro` [10-01-2019].
[18] `https://www.w3.org/TR/rdf11-concepts/#section-Graph-Literal`, `https://tools.ietf.org/html/bcp47#section-2.2.9` [11-01-2019]. – Note also that '@arab' is used to represent Arabic, although the ISO 639 code is 'ara', `https://iso639-3.sil.org/code/ara` [11-01-2019].

As little connected as N|uu, ǁ'Au, and Old French might ostensibly seem, they serve well to illustrate the problem: ISO 639 codes do not exist for N|uu, ǁ'Au, and the varieties of Old French. The atypicality of these languages highlights the relevance of the problem on a broader scale: more under-resourced, extinct or historical languages that are (currently) not included in the ISO 639 language code list will be published as LLOD.

3.1 N|uu and ǁ'Au

N‖ng is the name of a dialect cluster of the !Ui-Tuu language family (formerly referred to as Southern Khoisan), spoken over a geographically large area in the southern Kalahari Desert; N|uu is the Western variety of N‖ng, and ǁ'Au, the Eastern variety ([16, 11-17]; [33]; [5, 27]). Both dialects are near-extinct with two speakers for ǁ'Au and three speakers for N|uu as of 2013 (with the most fluent speaker of N|uu acting as a language teacher to young people); all N‖ng speakers use Afrikaans as their main language [5, 15-16]. Since the late 19th Century, linguists have collected data of Khoisan[19] languages: this data is sparse, heterogeneous and difficult to access with misclassified languages, inappropriate language names and insufficient metadata as examples of the challenges faced, in addition to the identity of diverse corpora in archival material hard to assess, both in relation to each other and to modern languages ([16, 5-8]; [5, 2]). To document the many Khoisan languages is a challenge and a desideratum at the same time: encoding data following the Linked Data paradigm will convert the data into a valuable resource, possibly giving way to linguistic reconstructions using computational methods, where standard linguistic methodologies have been unable to yield meaningful results [5, 1]. Making accessible and preserving this data will contribute significantly to the exploration of the cultural heritage of mankind, with the collective group of Khoisan speakers being one of the few remaining hunter-gatherer cultures worldwide and the oldest existing human group today, according to genetic studies [29, 379].

3.1.1 Existing language codes

In order to convert the linguistic data of N|uu and ǁ'Au resources, we need an appropriate means to denote the languages in an unambiguous way, i.e., language codes to label the modeled elements of the linguistic resources in RDF. A language code for N‖ng exists, i.e., ISO 639-3 'ngh'; this code is shared by both sub-languages N|uu and ǁ'Au.[20] However, according to the archival Khoisan 'doculects' discussed by [16, 16], the differences between the two language varieties are significant and, thus, explicit language codes for both ǁ'Au and N|uu are required.

Within MultiTree, a library of language relationships hosted by *The Linguist List*, the codes for N|uu and ǁ'Au are 'ngh-nuu' and 'ngh-aun' respectively.[21] Both are documented for 'Private Use', however their syntax does not meet the requirement defined by IETF's BCP 47, where the private use portion of the tag must be prepended with '`x-`' ([21]; [28, 4]). Furthermore, both the latter portions of MultiTree's codes, namely 'nuu' and 'aun', are pre-existing language codes, i.e., the former for the language Ngbundu (a language of the Congo area), and the latter for Molmo One (Papua New Guinea).[22] Despite the fact that

[19] The modern Khoisan languages are classified into three families and two isolates: the families Kx'a, !Ui-Tuu and Khoeid, and the isolates Hadza and Sandawe [5, 2].
[20] `https://iso639-3.sil.org/code/ngh` [29-12-2018].
[21] `http://www.multitree.org/codes/ngh-nuu`, `.../codes/ngh-aun` [20-06-2018].
[22] `https://www.iana.org/assignments/language-subtag-registry/language-subtag-registry` [29-12-2018].

the use of *privateuse* sub-tags is by definition by private agreement only (*cf.* Point 2.2.7.5 of BCP 47, [28, 18]), it is clear that the use of MultiTree's language tags 'ngh-nuu' and 'ngh-aun' may lead to inadvertent misinterpretation when included in a language tag.

For this reason, we consider the use of Glottolog, a comprehensive catalogue of the world's lesser-known languages maintained by the Max Planck Institute for the Science of Human History. Their catalogue «assigns a unique and stable identifier (the Glottocode) to (in principle) all languoids, i.e. all families, languages, and dialects», [17]. Glottolog registers the two languages N|u and ǁ'Au (as sub languages of N‖ng)[23] with the codes 'nuuu1242' and 'auni1243', respectively. However, as BCP 47 only allows for ISO 639 language codes in its *language* sub-tag, Glottolog is not recognized as a standard.

3.1.2 The use of *privateuse* sub-tag

In light of unambiguous language codes being available for the two Khoisan varieties, we propose to combine the ISO 639-3 code for the parent language N‖ng, i.e., 'ngh', with the *privateuse* sub-tag '`x-`' and the respective Glottocodes stated above.

The language tags for N|uu and ǁ'Au can then be defined accordingly:

- N|uu: `ngh-x-nuuu1242`
- ǁ'Au: `ngh-x-auni1243`

A lexical concept, which can be linked to one or more senses in lexical entries from different languages, can be modeled as follows:

```
1  @PREFIX skos: <http://www.w3.org/2004/02/skos/core#> .
2  @PREFIX dbr:  <http://dbpedia.org/resource/> .
3  ...
4
5  :concept/000000001 a skos:Concept , ontolex:LexicalConcept ;
6     skos:example              "The belly is fat"@en ;
7     skos:example              "ǁ'â he !qhûia."@ngh-x-nuuu1242 ;
8     ontolex:lexicalizedSense :en-n-belly#sense1 ;
9     ontolex:lexicalizedSense :ngh_x_nuuu1242-n-xa_belly#sense2 ;
10    ontolex:isConceptOf       dbr:Abdomen .
```

Where:

- Lines 6-7 show language-tagged strings, and line 7 the compiled language tag for N|uu.

3.2 Old French

Old French is the French spoken in the Middle Ages, and it can be more precisely defined as the umbrella term for the different Old French dialects[24] spoken in what is now France, parts of Belgium, England, Italy and the Holy Land. Its written resources date from 842 AD until c. 1350 AD (the border with Middle French) and its remarkable written tradition[25] serves to document its role as the most important vernacular of this time in Europe.

[23] `http://glottolog.org/resource/languoid/id/nuuu1241` [24-06-2018].
[24] The DEAF registers 30 varieties of Old French, Franco-Italian (a written, artificial language in the Middle Ages), and Judeo-French (sociolect), see Table 3, Appendix.
[25] Approx. 3,000 primary text sources transmitted within more than 10,000 manuscripts are registered by the *Complément bibliographique* of the DEAF, `http://www.deaf-page.de/bibl_neu.php` [07-01-2019].

3.2.1 Existing language codes

BCP 47's language tag offers a *variant* sub-tag that can be «used to indicate additional, well-recognized variations that define a language or its dialects that are not covered by other available subtags», where one or more variants can be used to form a language tag. Each of these variant sub-tags must be registered with IANA before use [28, 15]. Middle French is registered (ISO 639-3 code 'frm')[26] but no variants have been registered for Old French. IANA has registered Anglo-Norman (ISO 639-3 code 'xno'), but not as a sub-category of Old French, although it should be considered as such; the same applies to Zarphatic ('zrp': Judeo-French, spoken in the Middle Ages).

MultiTree lists Old French ('fro') and also the following child languages: Picard (ISO 639-3 code 'pcd'), Walloon ('wln'), and Zarphatic ('zrp'); Anglo-Norman ('xno') is not registered as a child language.[27] Although Walloon is registered as a child language of Old French, it is described as a living language; the same applies to Picard. Middle French is also registered as a child language of Old French, thus, following this logic, so should modern French. The hierarchization of Judeo-French (variety of Old French: sociolect) on the same level as Middle French (successor of Old French) and Picard / Walloon (modern dialects of the Picardy and Wallonia, respectively) conflates synchronic, diachronic, and geographical aspects.

Glottolog has assigned the identifier 'oldf1239' to Old French[28] but Glottolog does not register dialects of the medieval time period.[29] In addition to this flaw, Glottolog does not seem appropriate for the needs of linguists modeling data from the Romance languages, particularly with regard to old language stages. A closer look at Glottolog reveals major shortcomings in both the registration and the hierarchization of the Romance languages. E.g., Glottolog conflates diachronic and dialectal criteria within its hierarchies in several ways: Old French is registered (as a sub-entity of 'Oïl'[30]) at the same level as modern 'Central Oïl', Francoprovençalic (Romance language spoken in Eastern France), and Walloon. Following the hierarchy into the branches and sub-branches of 'Central Oïl' we find → Macro-French → Global French → French → a number of modern French dialects, but, also, Middle French and Anglo-Norman.[31] We deem necessary a thorough revision of the hierarchies, (re-)assembling both the dialects and regional varieties of modern French, and the historic stages of French.

3.2.2 Preliminary findings

The evaluation of language tags and language hierarchies in ISO 639, BCP 47, IANA, MultiTree, and Glottolog shows that the assignation of language codes to Old French dialects is not straightforward. At least for Anglo-Norman and Zarphatic, which we consider sub-categories of Old French, ISO 639-3 provides codes, i.e., 'xno' and 'zrp' respectively. These codes can be used for modeling lexemes and their graphical variants characterized as Anglo-Norman or Zarphatic. The following example for the Anglo-Norman noun *firbote*[32] illustrates this:

---

[26] The sub-tag 'frm-1606nict', `ftp://www.iana.org/assignments/lang-subtags-templates/1606nict.txt` [08-01-2019], does not depict a regional variety but the language documented by Jean Nicot in his *Thresor de la langue françoyse, tant ancienne que moderne*, Paris, from 1606.

[27] `http://www.multitree.org/codes/fro.html`; `.../pcd`; `.../wln`; `.../zrp`; `.../xno` [07-01-2019].

[28] `https://glottolog.org/resource/languoid/id/oldf1239` [07-01-2019].

[29] Old French is not available in the language collection of Ethnologue, as «ancient, classical, and long-extinct languages are not listed», `https://www.ethnologue.com/about/this-edition[29-12-2018]`.

[30] The term for the Romance varieties using an adaptation of the Vulgar Latin term *hoc ille* "this (is) it" as 'Yes'.

[31] More modern French dialects are found scattered in other sub-branches.

[32] Juridical term (in England) designating the right to take firewood from the land of a landlord, DEAF F 492,29, `https://deaf-server.adw.uni-heidelberg.de/lemme/firbote` [08-01-2019].

```
<firbote> a ontolex:LexicalEntry , ontolex:Word ;
  lexinfo:PartOfSpeech    lexinfo:Noun ;
  ontolex:canonicalForm   <firbote#form> .

<firbote#form> a ontolex:Form ;
  ontolex:writtenRep      "firbote"@xno .
```

3.2.3 The use of *privateuse* sub-tag

For the other Old French dialects and language varieties (see Table 3, Appendix), as language codes are not available, we again have to consider the use of BCP 47's *privateuse* sub-tag. E.g., a tag for the Old French variety spoken in Lorraine, a region in north-eastern France, could be defined as `fro-x-lorraine`. A simple example of an Old French word form characteristic of the Lorraine scripta is *feyvre*, a graphical variant of Old French *fevre* m.[33] This can be modeled as follows:

```
<fevre> a ontolex:LexicalEntry , ontolex:Word ;
   ontolex:canonicalForm <fevre#form_1> ;
   ontolex:otherForm     <fevre#form_2> .

# Old French standard form (lemma)
<fevre#form_1> a ontolex:Form ;
   ontolex:writtenRep    "fevre"@fro .

# graphical variant
<fevre#form_2> a ontolex:Form ;
   ontolex:writtenRep    "feyvre"@fro-x-lorraine .
```

3.2.4 Adding geographic information

The language tag can be further enriched by including geographic information, in line with established standards. There are several options available to us: (1) we could refer to the administrative region of France, (2) to the French *département*, or (3) use geographic coordinates. Both the administrative region and the *département* can be identified using the codes of the ISO 3166 standard for the administrative subdivisions of France.[34]

3.2.4.1 Administrative region and *département*

The area 'Lorraine' is part of the region Grand-Est (covering Alsace, Champagne, Ardenne, and Lorraine), thus the language tag can be defined as `fro-x-lorraine-FR-GES`.[35] However, the administrative region covers an area considerably larger than the geographic area of Lorraine, and thus does not map the area in question in a satisfying way. Another option would be to enrich the language tag by referring to the *département*, which would allow us to map the area more precisely.

Regarding options (1) and (2), the following concerns are raised:

[33] The smith, DEAF F 342,21, `https://deaf-server.adw.uni-heidelberg.de/lemme/fevre` [08-01-2019].
[34] `https://www.iso.org/obp/ui/#iso:code:3166:FR` [07-01-2019].
[35] *Ibid.*

(i) The administration of regions and *départements* is subject to change. As a consequence, the ISO 3166 codes are unstable, as evidenced by sub-divisions being allocated to new metropolitan regions in France as recently as 2016.[36]

(ii) The area in which an Old French dialect was spoken can embrace several modern regions, e.g., 'Nord-Est' and 'Sud-Ouest' (see Table 3, Appendix), or *départements*: e.g., contemporary Lorraine consists of not one but four *départements*, i.e., Meurthe-et-Moselle (ISO 3166-2:FR-54), Meuse (ISO 3166-2:FR-55), Moselle (ISO 3166-2:FR-57), and Vosges (ISO 3166-2:FR-88); the historical region also comprises the contemporary *département* Haute-Marne (ISO 3166-2:FR-52). As a result, either more than one region may need to be included in the sub-tag, indicating (imprecisely) the geographical boundary in which the dialect was spoken, or the RDF triples must be manifolded: when modeling a lexeme or a graphical variant of a lexeme characterized as Lorraine, e.g., within the data of the DEAF dictionary, the inclusion of the codes for the *départements* into the language tag requires duplicating the RDF triples, thus creating somewhat unwieldy data.

(iii) The boundaries of the regions are modern-day boundaries which may not necessarily align to the boundaries of a previous time. This leads to a dissatisfying mapping of said area.

3.2.4.2 Geographic coordinates

As a third option, we consider the inclusion of geographic coordinates in the language tag. To do this, we map the (approximate) geographic distribution of 'Lorraine' to coordinates, assuming that the last coordinate is the same as the first coordinate, and the coordinates are ordered in a counterclockwise direction, thus creating a polygon shape [25]. Each coordinate can be compressed using Geohash, a system for encoding geographic coordinates into a base32 string, which would also format each latitude and longitude value in a syntax acceptable for BCP 47.[37] As precision down to the nearest meter is not necessary, the Geohash length could be limited to five characters,[38] rendering the coordinate in an approximate area that is $\leq$ 4.89 x 4.89 kilometers.[39]

As using the geographic coordinates to map the modern-day distribution of 'Lorraine' would lead to the same dissatisfying result (*cf.* 3.2.4.1), we draw on a map of the *Französisches Etymologisches Wörterbuch* – FEW [34] that includes historical information, see Fig. 1.[40]

To derive the geographic coordinates for the old dialect of 'Lorraine', we take this map as a substratum and the result is the following:
(4.91473,49.62686), (4.6696405,48.0428789), (5.59192,47.6435), (6.858446002006532, 47.883257283545234), (7.2386756,48.4086571), (5.81263,49.72584), (4.91473,49.62686)[41]

---

[36] `https://www.iso.org/obp/ui/#iso:code:3166:FR` [29-12-2018].

[37] Geohash 2018, `https://en.wikipedia.org/wiki/Geohash` [31-12-2018]; Geo-shape datatype 2018, `https://www.elastic.co/guide/en/elasticsearch/reference/current/geo-shape.html` [31-12-2018].

[38] Or less, depending on the extent of the geographical distribution of the dialect being mapped.

[39] `https://www.movable-type.co.uk/scripts/geohash.html` [31-12-2018].

[40] In the possession of the editorial office of the DEAF is a 40-year-old, battered copy of the map of France that is included in the *Beiheft* of the FEW. This copy contains the boundaries of the areas where the Old French dialects were spoken, sketched in by hand (in yellow) by Frankwalt Möhren, co-founder of the DEAF (and, also, valuable notes and comments, e.g., the indication 'Orval: Bier!': the Abbey of Orval in Villers-devant-Orval is the home of the famous top-fermented beer 'Orval').

[41] Ordering is latitude then longitude.

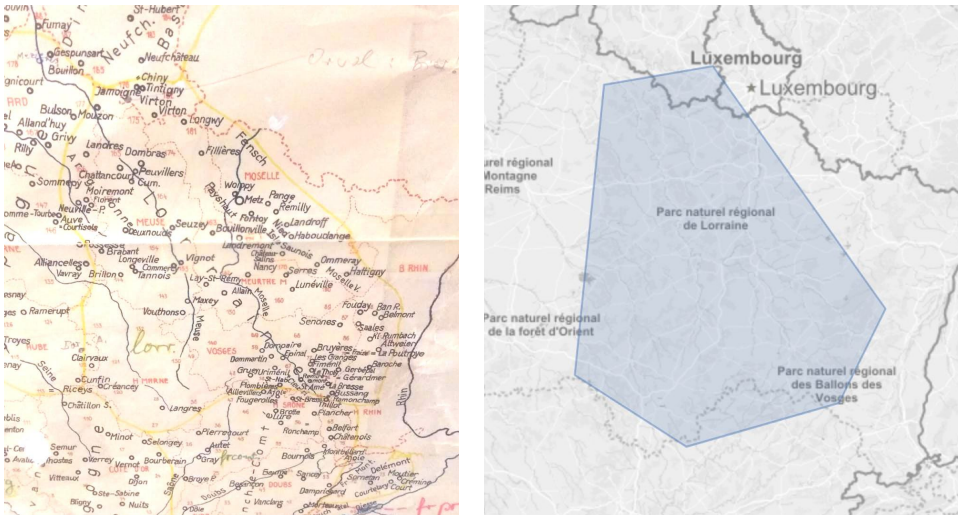


Figure 1 'Old Lorraine' area: Extract of the map of the FEW (left), mapped using geographic coordinates (right).

Each longitude and latitude coordinate can be converted to a Geohash, to a precision of five characters: `t0g7c, t0f4t, t0czu, t14p1, t163j, t1535, t0g7c`.

As the last coordinate is the same as the first coordinate, the last one can be excluded, and as only alphanumeric characters and hyphens are allowed by BCP 47, every Geohash, with the exception of the first one, is prepended with '`--`' to serve as an internal delimiter; the language code for the language, dialect and region can thus be presented as follows: `fro-x-lorraine-t0g7c--t0f4t--t0czu--t14p1--t163j--t1535`.

The use of a historical map as a source of information to enrich a language tag with geographic coordinates, as demonstrated for medieval Lorraine, seems very promising to us regarding our aim: the unambiguous and historically-correct tagging of languages.

A further possibility is to include the period of time within the language tag, e.g., `fro-x-lorraine-t0g7c--t0f4t--t0czu--t14p1--t163j--t1535-850AD--1350AD`, where `850AD--1350AD` depicts the time range.[42]

BCP 47 specifies the maximum length of a sub-tag to be of eight characters (+ two for '`x-`', see [28, 6]). However, numerous examples of the *privateuse* sub-tags exceed this maximum length [28, 56,81]. Thus, we conclude that there is not an upper limit to the length of the *privateuse* sub-tag, except that pertaining to buffer overflow [28, 63,71-72].

4 Discussion

The examples, N|uu and ǁ'Au, and Old French, demonstrate that there is not a single, encompassing solution that can be applied to all languages. For each of the three languages, a custom approach, in conjunction with the *privateuse* sub-tag from BCP 47's language tag, has had to be adopted. However, with each example, a tentative pattern for the *privateuse* sub-tag has emerged: each part within the *privateuse* sub-tag can be assigned to a category, as listed in Table 1, and the *privateuse* sub-tag can consist of one or more parts.

[42] In the case of Old French, this seems dispensable since the code 'fro' contains this information, however it could be valuable when identifying a language where the geographical distribution changes significantly.

Table 1 The categorization of parts in a *privateuse* sub-tag.

Part	Description
language	A language, dialect or pidgin not in ISO 639
otherlect	An ethnolect, sociolect, or idiolect
timeperiod	If not modern-day; not equivalent to the time period specified by the language code
region	A geographic, politic or administrative region

Using the categories identified in Table 1, we thus propose the following pattern for the *privateuse* sub-tag of a language tag, with each part separated by a '-':

`x-language-otherlect-timeperiod-region`

Within BCP 47, the format of the language tag has been designed such that each sub-tag can be identified on the basis of its length, position in the tag, and its content, and each sub-tag is typically a code from an ISO standard or registry [28, 8]. However, this requirement can be limiting and inflexible. In order to identify each part in the *privateuse* sub-tag pattern, we propose prepending each part with a key consisting of 2 digits, from 0 - 9, with the first digit, Key 1, indicating the category, and the second digit, Key 2, indicating the content in relation to Key 1, as shown in Table 2. This way, each part can be of variable length, thus allowing for greater flexibility. For example, a part that is categorized as *language* can be prepended with '10', where '1' indicates that it is *language* and '0' indicates that the language is user-defined information. The tags can, thus, be rewritten as follows:

- N|uu dialect: `ngh-x-01nuuu1242`
- ǁ'Au dialect: `ngh-x-01auni1243`
- Old French, Lorraine dialect:
  `fro-x-00lorraine-30t0g7c--t0f4t--t0czu--t14p1--t163j--t1535`

Table 2 The key for each part of the *privateuse* sub-tag.

Part	Key 1	Key 2
language	0	0 = User-defined 1 = Glottocode
otherlect	1	0 = User-defined 1 = Glottocode
timeperiod	2	0 = one year only, BC 1 = one year only, AD 2 = start:BC - end:BC 3 = start:BC - end:AD 4 = start:AD - end:AD
region	3	0 = Geohashed latitude and longitude coordinates – polygon 1 = Geohashed latitude and longitude coordinates – point only 2 = URI to GeoJSON-LD 3 = Code from ISO 3166

The interpretation of a language tag which contains multiple sub-tags can be obscure and requires human inspection. By (1) categorizing the *privateuse* sub-tag into parts, then (2) defining a key for each part, and (3) defining rules for each key, it not only allows for more accurate interpretation, by both human and machine, but it can also lead to increased shared agreement for a compiled language tag.

5 Conclusion and Future Work

In this paper, we have discussed the shortcomings of language tags in the context of modeling data from lesser-known languages as LD. For two under-resourced language varieties and one historical language stage we have proposed solutions using the *privateuse* sub-tag, with the addition of geographic information. This can improve a language tag so that it reflects the diachronic, synchronic and dialectal aspects of the language in question.

The proposed rule-based pattern for the *privateuse* sub-tag is not intended to be used in place of other sub-tags in the language tag, nor is it intended to replace the work of existing standards and bodies. The W3C Internationalization (i18n) Interest Group[43] serves to connect a large group of people on the topic of internationalization on the Web. The authors intend to contribute to the discussions of the group, submitting the proposals outlined in this paper for further feedback. Also, the authors and C. Maria Keet propose MoLA, a **Mo**del for **L**anguage **A**nnotation (`https://ontology.londisizwe.org/mola`) [14]. MoLA has been developed to provide a vocabulary for language annotation in RDF, which enables custom language tags to be defined, and for said language tags to be associated with both a time period and region.

Defining a pattern for the *privateuse* sub-tag can lead to discussions which can improve the next iteration of BCP 47, as well as to increased interoperability within the context of LLOD so as to render language identification more accurate. This in turn can lead to shared agreement between lexical resources and to re-use, an important notion in a multilingual Semantic Web.

References

1. K. Baldinger. *Dictionnaire étymologique de l'ancien français – DEAF*. Presses de L'Université Laval / Niemeyer / De Gruyter, Québec/Tübingen/Berlin, since 1971. [Continued by Frankwalt Möhren, and Thomas Städtler; DEAF*él*: `https://deaf-server.adw.uni-heidelberg.de`].
2. A. Bellandi, E. Giovannetti, and A. Weingart. Multilingual and Multiword Phenomena in a lemon Old Occitan Medico-Botanical Lexicon. *Information*, 9 (3), 52, 2018.
3. T. Berners-Lee. *Linked Data*. World Wide Web Consortium, 2006.
4. Ch. Bizer, T. Heath, and T. Berners-Lee. Linked Data – The Story So Far. *International Journal on Semantic Web and Information Systems*, 5:1–22, 2009.
5. M. Brenzinger. The twelve modern Khoisan languages. In A. Witzlack-Makarevich and M. Ernszt, editors, *Khoisan Languages and Linguistics: Proceedings of the 3rd International Symposium July 6-10, 2008, Riezlern / Kleinwalsertal*, pages 1–32. Köppe Verlag, 2008.
6. Ch. Chiarcos, J. McCrae, Ph. Cimiano, and Ch. Fellbaum. Towards Open Data for Linguistics: Lexical Linked Data. In A. Oltramari et al., editor, *New Trends of Research in Ontologies and Lexical Resources: Ideas, Projects, Systems*, pages 7–25. Springer, Berlin, Heidelberg, 2013.
7. Ch. Chiarcos and M. Sukhareva. Linking Etymological Databases. A Case Study in Germanic. In *3rd Workshop on Linked Data in Linguistics: Multilingual Knowledge Resources and Natural Language Processing*, page 41, 2014.
8. P. Cimiano, J.P. McCrae, and P. Buitelaar. Lexicon model for ontologies: community report, 10 May 2016. Ontology-Lexicon Community Group under the W3C Community Final Specification Agreement (FSA), 2016. URL: `https://www.w3.org/2016/05/ontolex/`.
9. R. Cyganiak, D. Wood, and M. Lanthaler. RDF 1.1. concepts and abstract syntax: W3C recommendation 25 February 2014, 2014. URL: `https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/`.

[43] `https://www.w3.org/International/ig/Overview` [15-03-2019].

10 Gerard de Melo. Lexvo.org: Language-Related Information for the Linguistic Linked Data Cloud. *Semantic Web*, 6(4):393–400, August 2015.

11 Th. Declerck, E. Wandl-Vogt, and K. Mörth. Towards a Pan European Lexicography by Means of Linked (Open) Data. In I. Kosem et. al., editor, *Electronic Lexicography in the 21st Century: Linking Lexical Data in the Digital Age. Proceedings of the eLex 2015 Conference, 11-13 August 2015, Herstmonceux Castle, United Kingdom*, pages 342–355. Trojina, Institute for Applied Slovene Studies/Lexical Computing Ltd., 2015.

12 International Organization for Standardization. Language codes – ISO 639. URL: `https://www.iso.org/iso-639-language-codes.html`.

13 F. Gillis-Webber. Conversion of the English-Xhosa Dictionary for Nurses to a linguistic linked data framework. *Information*, 9(11), 2018. `doi:10.3390/info9110274`.

14 F. Gillis-Webber, S. Tittel, and C. M. Keet. A Model for Language Annotations on the Web, 2019. (submitted).

15 J. Gracia, M. Villegas, A. Gómez-Pérez, and N. Bel. The Apertium Bilingual Dictionaries on the Web of Data. In *Semantic Web – Interoperability, Usability, Applicability*, pages 1–10. IOS Press, 2017.

16 R. Güldermann. Towards casting a wider net over N‖ng: chances and challenges of archival Khoisan resources, 2014. URL: `https://www.iaaw.hu-berlin.de/de/region/afrika/afrika/linguistik/mitarbeiter/1683070/dokumente/2014-03-cape-town-nng-h`.

17 H. Hammarström, R. Forkel, and M. Haspelmath. Glottolog 3.3., 2018. accesssed 21-02-2019.

18 SIL International. ISO 639-3: Relationship between ISO 639-3 and the other parts of ISO 639, 2017. URL: `https://iso639-3.sil.org/about/relationships`.

19 SIL International. ISO 639-3: Scope of denotation for language identifiers, 2017. URL: `https://iso639-3.sil.org/about/scope`.

20 SIL International. ISO 639-3: Types of individual languages, 2017. URL: `https://iso639-3.sil.org/about/types`.

21 R. Ishida. Language Tags in HTML and XML, 2014. URL: `https://www.w3.org/International/articles/language-tags/index.en`.

22 F. Khan, J.E. Díaz-Vera, and M. Monachini. The Representation of an Old English Emotion Lexicon as Linked Open Data. In John P. McCrae et al., editor, *Proceedings of the LREC 2016 Workshop "LDL 2016 – 5th Workshop on Linked Data in Linguistics: Managing, Building and Using Linked Language Resources", 24 May 2016 – Portorož, Slovenia*, pages 73–76, 2016.

23 G. Köbler. *Wörterbuch des althochdeutschen Sprachschatzes*. Schöningh, Paderborn, 1993.

24 L. Lezcano, S. Sánchez-Alonso, and A. Roa-Valverde. A Survey on the Exchange of Linguistic Resources. *Program*, 47,3:263–281, 2013.

25 J. Lieberman, R. Singh, and Ch. Goad. W3C geospatial vocabulary: W3C incubator group report 23 October 2007, 2007.

26 J.P. McCrae, J. Bosque-Gil, J. Gracia, P. Buitelaar, and P. Cimiano. The OntoLex-Lemon model: Development and Applications. In *Proceedings of ELEX 2017: Lexicography from Scratch. September 2017*, pages 19–21, 2017.

27 S. Moran and M. Brümmer. Lemon-aid: Using Lemon to Aid Quantitative Historical Linguistic Analysis. In Ch. Chiarcos et al., editor, *Proceedings of the 2nd Workshop on Linked Data in Linguistics (LDL-2013), Pisa, September 2013*, pages 28–33. Ass. for Comp. Linguistics, 2013.

28 A. Phillips and M. Davis. Tags for Identifiying Languages. *BCP*, 47, 2009.

29 C.M. Schlebusch, P. Skoglund, and P. Sjödin et al. Genomic Variation in Seven Khoe-San Groups Reveals Adaptation and Complex African History. *Science*, 338(6105):374–379, 2012.

30 S. Shah and M. Brenzinger. *Ouma Geelmeid ke kx'u ‖xa‖xa N|uu*. Centre for African Language Diversity, University of Cape Town, Cape Town, 2016.

31 S. Tittel, H. Bermúdez-Sabel, and Ch. Chiarcos. Using RDFa to Link Text and Dictionary Data for Medieval French. In J.P. McCrae et al., editor, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). 6th Workshop on Linked Data in Linguistics (LDL-2018), Miyazaki, Japan, 2018*, pages 30–38, Paris (ELRA), 2018.

32 S. Tittel and Ch. Chiarcos. Historical Lexicography of Old French and Linked Open Data: Transforming the Resources of the *Dictionnaire étymologique de l'ancien français* with OntoLex-Lemon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). GLOBALEX Workshop (GLOBALEX-2018), Miyazaki, Japan, 2018*, pages 58–66, Paris (ELRA), 2018.

33 M. Van Der Merwe. Giving breath to a dying history, 2015. URL: `https://www.dailymaverick.co.za/article/2015-01-23-giving-breath-to-a-dying-history/#.Wyvou9WFMsk`.

34 W. von Wartburg. *Französisches Etymologisches Wörterbuch. Eine darstellung des galloromanischen sprachschatzes – FEW*. ATILF, since 1922. [Continued by O. Jänicke, C.T. Gossen, J.-P. Chambon, J.-P. Chauveau, and Yan Greub].

35 D. Wood, M. Zaidman, L. Ruth, and M. Hausenblas. *Linked data: structured data on the web*. Manning Publications Co., New York, 2014.

A Old French dialects

Table 3 List of Old French dialects (described in French) registered by the DEAF.

Abbrev.	Language	Abbrev.	Language
afr.	ancien français	saint.	saintongeais
mfr.	moyen français	tour.	tourangeau
fr. du 16^{e}s.	français du 16^{e} siècle	orl.	orléanais
fr.dial.	français dialectal	bourb.	bourbonnais
frc.	francien (français de l'Ile de France)	bourg.	bourguignon
pic.	picard	lyon.	lyonnais
flandr.	français de la Flandre française	frcomt.	franc-comtois
hain.	hennuyer	francoit.	franco-italien
art.	artésien	Nord-Est	
wall.	wallon	Nord	
liég.	liégeois	Nord-Ouest	
champ.	champenois	Ouest	
lorr.	lorrain	Sud-Ouest	
norm.	normand	Centre	
agn.	anglo-normand	Est	
hbret.	haut-breton	Sud-Est	
ang.	angevin	Terre Sainte	
poit.	poitevin	judéofr.	judéofrançais

Functional Representation of Technical Artefacts in Ontology-Terminology Models

Laura Giacomini
University of Hildesheim, Germany
University of Heidelberg, Germany
laura.giacomini@uni-hildesheim.de

Abstract

The ontological coverage of technical artefacts in terminography should take into account a functional representation of conceptual information. We present a model for a function-based description which enables direct interfacing of ontological properties and terminology, and which was developed in the context of a project on term variation in technical texts. Starting from related research in the field of knowledge engineering, we introduce the components of the ontological function macrocategory and discuss the implementation of the model in *lemon*.

2012 ACM Subject Classification Information systems → Ontologies

Keywords and phrases terminology, ontology, technical artefact, function model, semantic web, lemon

Digital Object Identifier 10.4230/OASIcs.LDK.2019.5

Category Extended Abstract

1 Introduction

In the framework of a larger terminology project carried out at Hildesheim University, we have been designing a formal ontology of technical artefacts relevant to the field of thermal insulation in buildings, subsequently using the ontology as a knowledge base for a technical e-dictionary. We have thus had the opportunity to reflect on the requirements that such an ontology must meet in order to represent in an exact, coherent and replicable way conceptual information regarding technical artefacts, complying at the same time with terminological description. In this contribution, we would like to report on preliminary work concerning the functional representation of technical artefacts within an ontology-terminology model.

Our report focuses on technical artefacts as one of the most prominent types of extra-linguistic objects from the point of view of terminology, terminography, and specialised translation. Semantic Web-oriented studies are making steady progress in the field of formal ontologies, especially with regard to ontology-related semantic deep learning tasks (cf. Gromann/ Declerck 2018 [7]), ontology learning techniques (cf. Asim et al. 2018 [1] for an overview), and the development of models for lexica representation (e.g. *lemon*, McCrae et al. 2011 [10]). However, little has been done so far to systematically describe the typical characteristics of certain classes of ontological objects. Some interesting ideas about the specific characteristics of technical artefacts emerge from studies in the field of domain knowledge engineering, in which particular attention is paid to functional aspects (cf. Section 3). We have taken this as our starting point for developing a model for terminology information systems.

This contribution shows how a function-based ontological description can be integrated in terminographic resources dealing with technical artefacts. After introducing function as a macrocategory in our ontological model (Section 2), we discuss the typical terminological implications of a function-based approach to knowledge engineering (Section 3). Next we present our model for a functional representation of technical artefacts (tested on texts


© Laura Giacomini;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 5; pp. 5:1–5:6

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

concerning insulation products and power tools) and discuss its implementation in *lemon*, the lexicon model for ontologies (Section 4). We finally draw some conclusions on the accomplished work and its challenges and provide information on future work.

2 Function as an ontological macrocategory

By technical artefact we mean a physical object with technical features commercialised and used as a finished product. As pointed out in Giacomini (2018 [6]), technical artefacts can be appropriately described in terms of MATTER, FORM AND FUNCTION, three ontological macro-categories drawing on the Aristotelian description of inanimate objects, to which specific properties of an artefact can be linked. Functional knowledge plays a particularly crucial role in our cognitive perception of the artefact and is closely related to design intentions (Motta et al. 2011: 99 [12]). The dual nature of technical artefacts as the combination of structural and intentional conceptualisations has been highlighted in a number of recent studies in philosophy of science (cf., among others, Vermaas/ Houkes 2006 [14], Houkes/ Meijers 2006 [9] and Motta et al. 2011 [12]).

Borgo et al. (2016: 242 [2]) observe that several definitions of function have been formulated in engineering design, philosophy and ontology research. The unified definition of function for biological systems and technical artefacts proposed by Mizoguchi et al. (2016) [11] for a foundation ontology best suits our terminological purposes. According to the authors, different types of contexts identify different types of functions (here: functional roles): "In systemic contexts, the functional role is given by the systemic context where the appropriateness of its goal is determined with respect to the (goal provided by the) selected behavior of the overall system, which has the functional object as component. In the case of design contexts, the functional role is determined by the designer's intention. Finally, in the case of the use context, the determination is given by the user's intention" (ibid.: 141). Moreover, the notion of context in which the functions of an artefact are embedded legitimises a frame-based semantic approach to technical terminology as presented in (Giacomini 2018 [6], Faber 2012 [3]), with frames (Fillmore 2006 [4]) as complex cognitive structures identified against the background of a specified context.

3 Function models in knowledge engineering and their terminological implications

Some of the function models proposed in the field of knowledge engineering are designed to be integrated into upper ontologies, and not for interfacing with a terminology layer of a terminology resource. Others, however, describe conceptual elements of a domain ontology and can therefore be used for immediate classification of terminological elements. This is for example the case of the Reconciled Functional Basis (RFB) model presented by Hirtz et al. (2002 [8]) and aimed at supporting taxonomical modelling of engineering functions (e.g. isolate, move, associate) and flows (e.g. pressure, energy, velocity). In Reconciled Functional Basis, function and flow primary classes increase in specification at the secondary and tertiary levels and are associated to specific terms (typically verbs for functions and nouns for flows), e.g. in the following function set (Figure 1):

This model has successfully been applied to engineering design tasks (for instance to the building of an engineering-to-biology thesaurus, cf. Nagel et al. 2010 [13]). Its main drawback, however, is its potential ambiguity from the point of view of natural language, i.e. the semantic ambiguity of terms simultaneously attributed to more than one function

Class (Primary)	*Secondary*	*Tertiary*	*Correspondents*
Branch	Separate		Isolate, sever, disjoin
		Divide	Detach, *isolate*, release, sort, split, disconnect, subtract
		Extract	Refine, filter, purify, percolate, strain, *clear*
		Remove	Cut, drill, lathe, polish, sand
	Distribute		Diffuse, dispel, disperse, dissipate, diverge, scatter
Channel	Import		Form entrance, *allow*, input, *capture*
	Export		Dispose, eject, *emit*, empty, *remove*, destroy, eliminate
	Transfer		Carry, deliver
		Transport	Advance, lift, move
		Transmit	Conduct, convey
	Guide		Direct, shift, steer, straighten, switch
		Translate	Move, relocate
		Rotate	Spin, turn
		Allow DOF	*Constrain*, unfasten, unlock
Connect	Couple		Associate, connect
		Join	Assemble, fasten
		Link	Attach
	Mix		Add, blend, coalesce, combine, pack

Figure 1 Example of engineering functions in Reconciled Functional Basis (Hirtz et al. 2002).

or flow class, as well as the non-exhaustiveness of terminological coverage. The example of the RFB model shows that, for obtaining a coherent treatment of natural language, the ontology structure and contents should not condition the terminological component of the model. Instead of a strict top-down method, a terminology-oriented approach to ontology design should also take advantage from corpus-based terminological analysis to grasp relevant ontological aspects (combined top-down and bottom-up approach). In the next section, examples will be shown for the representation of function-related ontological properties by relying on domain corpus data concerning technical artefacts.

4 Normal function and functional properties of technical artefacts

Functional representation

As mentioned in the previous section, we use a corpus-based method to derive from specialised texts relevant information for the compilation of the domain ontology. In the context of the main study, terms and term relations were automatically extracted from a corpus of German technical texts and associated with elements of a previously defined frame "Functionality of the technical artefact" (for details of the extraction process, cf. Giacomini 2017 [5]). The syntactic and semantic behaviour of artefact-related terms in texts revealed a range of conceptual features that are crucial to knowledge representation. The validation experiments we later carried out not only in the field of thermal insulation but also in other technical subfields (power tools and semiconductor devices), show that a technical artefact usually has a *normal function* NF (e.g. a function conforming to a norm, also *systemic function* according to Mizoguchi et al. 2016 [11], or *use plan* according to Vermaas/ Houkes 2006 [14]): a thermal insulation product, for instance, is normally intended for thermally insulating a part of a building. The context in which the normal function of an artefact is performed can be interpreted as the sum of different conceptual constituents, which we call *functional properties* (FP):

(a) FP_project: Activity required of a technical artefact (TA) in its normal function.
(b) FP_location: Location in which a TA is used in its normal function.
(c) FP_patient: Object on which a TA operates in its normal function.
(d) FP_patient stuff: Material of FP_patient requiring the use of a certain TA to accomplish a certain function.

(e) FP_preparation: Process of making a TA ready for operation.
(f) FP_placement: Process of establishing a (physical) contact between TA and FP_patient before its FP_operation.
(g) FP_operation: Way in which a TA is used and operates in its normal function, typically procedural information or special techniques.
(h) FP_instrument: Medium by which FP_preparation, FP_placement, or FP_operation can be carried out on a TA used in its normal function.
(i) FP_agent: Performer of an action in which a TA is used in its normal function.

Table 1 illustrates the functional representation of two technical artefacts, an insulation roll and a circular saw. Here, we have manually attributed textual data (single-word terms, multi-word terms, and sentences) retrieved from online specialised texts in English to the different functional properties (sources: `https://www.tooled-up.com/artwork/ProdPDF`, `https://www.hilti.be`, `https://www.insulationsuperstore.co.uk`) . Depending on the artefact, some properties may be indicated as non-relevant (n.r.) for the given corpus contexts.

Table 1 Normal function (NF) and Functional properties (FP) of technical artefacts.

Functional representation	**Insulation roll**	**Circular saw**
NF	thermally insulate	saw
FP_project	to insulate a roof	to cut a wooden plank
FP_location	building	n.r.
FP_patient	roof	plank
FP_patient stuff	wood	wood
FP_preparation	to roll out	to switch on
FP_placement	a) push between the rafters, b) all joints must be taped	a) position the saw on the guide rail, b) position the saw against the workpiece
FP_operation	n.r. (not explicitly expressed in the available data set)	a) rotation, b) guide the circular saw along the cutting line, c) carry out a trial cut
FP_instrument	tape	teethed blade
FP_agent	craftsman	craftsman

The combination of normal function and functional properties lays the foundations for a functional representation of a technical artefact in a formal ontology. In the next future, we intend to explore the possibility of automatically processing our data sets to obtain function-related information from technical texts both in German and in English.

Integration in *lemon*

For terminographical purposes, this functional representation could be embedded in the lexicon model for ontologies (*lemon*), which was developed for enriching ontologies with natural language data (`https://www.w3.org/2016/05/ontolex`). Our present task is to test the extent to which our functional terminology description can be supported by the *lemon* model, specifically by the Ontolex module, and to propose the inclusion of some necessary components. The main benefit of this is the possibility of expanding the conceptual coverage of technical terminology, especially of multi-word terms (e.g. *thermally insulate*) and longer text segments (e.g. *position the saw on the guide rail*), in *lemon*.

Ontolex employs the rdfs:label to express lexicalisations. Semantic properties, in particular, are represented by means of the denote property as well as the sense and reference properties, which link lexical entries (and their lexical senses) to ontology entities. Given a Lexical Entry *building* with the Lexical Sense "building" in the domain of thermal insulation, we may use the reference property to relate this sense to the corresponding ontological predicate:

```
:lex_building a ontolex:LexicalEntry;
    ontolex:canonicalForm :form_building;
    ontolex:sense :building_sense.

:form_building ontolex:writtenRep "building"@en.

:building_sense a ontolex:LexicalSense;
    ontolex:reference <http://dbpedia.org/ontology/building>.
```

In addition to this, an indirect link of the Lexical Sense "building" to an Ontology Entity can take place via the Lexical Concept class, which is relevant for our functional representation of technical artefacts. In order to allow for a functional representation of concepts, in fact, we should specify normal functions and functional properties as *lemon* object properties. We propose, for instance, the integration of these properties at the Lexical Concept level. This means that, for the given example, the Lexical Concept "building" should be represented as rdfs:label FP_location property of the Lexical Concept "insulation roll":

```
:insulation_roll a ontolex:LexicalEntry;
    ontolex:sense :insulation_roll_sense;
    ontolex:evokes :insulation_roll.
:building a ontolex:LexicalConcept;
    ontolex:FP_location :insulation_roll.
```

The same can be done of the other functional properties and of the normal function of a technical artefact. Some (structural and conceptual) challenges concern, for instance, the exact location in which functional labels should be included into Ontolex, i.e. possibly at the Lexical Sense level as well. Moreover, in order to make the most of the potential of the functional model in technical terminology, lexical representation should take into account not only Lexical Entries in the form of single-word and multi-word terms, but also other relevant textual patterns (e.g. *push between the rafters* in Table 1) referring to functional features of artefacts. Finding a suitable solution to these challenges is our objective in the near future.

5 Conclusions and future work

Our research is aimed at finding helpful solutions for interfacing ontology and terminology in terminographic resources dealing with technical artefacts. At the moment, we are verifying the feasibility of a formalisation of our functional model in *lemon* by adding normal functions and functional properties to the Ontolex module in the form of new object properties. The current experimental results are rather promising, as they show a good flexibility of the functional model in adapting to different technical domains. The work ahead will also involve evaluation of the proposed functional model as well as the implementation of a frame-based layer to further enrich the semantic description and cross-referencing of terms with context-dependent information.

References

1 Muhammad N. Asim, Muhammad Wasim, Muhammad U. G. Khan, Waqar Mahmood, and Hafiza M. Abbasi. A survey of ontology learning techniques and applications. *Database*, 2018(1), 2018.

2 Stefano Borgo, Riichiro Mizoguchi, and Yoshinobu Kitamura. Formalizing and Adapting a General Function Module for Foundational Ontologies. In *FOIS*, pages 241–254, 2016.

3 Pamela Faber. *A cognitive linguistics view of terminology and specialized language*, volume 20. Walter de Gruyter, 2012.

4 Charles J. Fillmore. Frame semantics. *Cognitive linguistics: Basic readings*, 34:373–400, 2006.

5 Laura Giacomini. An Ontology-terminology Model for Designing Technical e-dictionaries: Formalisation and Presentation of Variational Data. In *Electronic lexicography in the 21st century: Proceedings of eLex 2017 conference*, pages 110–123. Lexical Computing, 2017.

6 Laura Giacomini. Frame-based Lexicography: Presenting Multiword Terms in a Technical E-dictionary. In *Proceedings of the XVIII EURALEX International Congress*, 2018.

7 Dagmar Gromann and Thierry Declerck. Comparing Pretrained Multilingual Word Embeddings on an Ontology Alignment Task. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

8 Julie Hirtz, Robert B. Stone, Daniel A. McAdams, Simon Szykman, and Kristin L. Wood. A functional basis for engineering design: reconciling and evolving previous efforts. *Research in engineering Design*, 13(2):65–82, 2002.

9 Wybo Houkes and Anthonie Meijers. The ontology of artefacts: the hard problem. *Studies in history and philosophy of science part A*, 37(1):118–131, 2006.

10 John McCrae, Dennis Spohr, and Philipp Cimiano. Linking lexical resources and ontologies on the semantic web with lemon. In *Extended Semantic Web Conference*, pages 245–259. Springer, 2011.

11 Riichiro Mizoguchi, Yoshinobu Kitamura, and Stefano Borgo. A unifying definition for artifact and biological functions. *Applied Ontology*, 11(2):129–154, 2016.

12 E. Motta, N. Shadbolt, and A. Stutt. Engineering Knowledge in the Age of the Semantic Web: Proceedings. In *14th International Conference, EKAW*, pages 5–8, 2004.

13 Jacquelyn K.S. Nagel, Robert B. Stone, and Daniel A. McAdams. An engineering-to-biology thesaurus for engineering design. In *ASME 2010 international design engineering technical conferences and computers and information in engineering conference*, pages 117–128. American Society of Mechanical Engineers, 2010.

14 Pieter E. Vermaas and Wybo Houkes. Technical functions: a drawbridge between the intentional and structural natures of technical artefacts. *Studies in History and Philosophy of Science Part A*, 37(1):5–18, 2006.

Comparison of Different Orthographies for Machine Translation of Under-Resourced Dravidian Languages

Bharathi Raja Chakravarthi
Insight Centre for Data Analytics, Data Science Institute, National University of Ireland Galway, IDA Business Park, Lower Dangan, Galway, Ireland
https://bharathichezhiyan.github.io/bharathiraja/
bharathi.raja@insight-centre.org

Mihael Arcan
Insight Centre for Data Analytics, Data Science Institute, National University of Ireland Galway, IDA Business Park, Lower Dangan, Galway, Ireland
michal.arcan@insight-centre.org

John P. McCrae
Insight Centre for Data Analytics, Data Science Institute, National University of Ireland Galway, IDA Business Park, Lower Dangan, Galway, Ireland
https://john.mccr.ae/
john.mccrae@insight-centre.org

Abstract

Under-resourced languages are a significant challenge for statistical approaches to machine translation, and recently it has been shown that the usage of training data from closely-related languages can improve machine translation quality of these languages. While languages within the same language family share many properties, many under-resourced languages are written in their own native script, which makes taking advantage of these language similarities difficult. In this paper, we propose to alleviate the problem of different scripts by transcribing the native script into common representation i.e. the Latin script or the International Phonetic Alphabet (IPA). In particular, we compare the difference between coarse-grained transliteration to the Latin script and fine-grained IPA transliteration. We performed experiments on the language pairs English-Tamil, English-Telugu, and English-Kannada translation task. Our results show improvements in terms of the BLEU, METEOR and chrF scores from transliteration and we find that the transliteration into the Latin script outperforms the fine-grained IPA transcription.

2012 ACM Subject Classification Computing methodologies → Machine translation

Keywords and phrases Under-resourced languages, Machine translation, Dravidian languages, Phonetic transcription, Transliteration, International Phonetic Alphabet, IPA, Multilingual machine translation, Multilingual data

Digital Object Identifier 10.4230/OASIcs.LDK.2019.6

Funding This work was supported in part by the H2020 project "ELEXIS" with Grant Agreement number 731015 and by Science Foundation Ireland under Grant Number SFI/12/RC/2289 (Insight).

1 Introduction

Worldwide, there are around 7,000 languages [1, 18], however, most of the machine-readable data and natural language applications are available in very few popular languages, such as Chinese, English, French, or German. For other languages resources are scarcely available and for some languages not at all. Some examples of these languages do not even have a writing system [28, 24, 2], or are not encoded in major schemes such as Unicode. The languages addressed in this work, i.e. Tamil, Telugu, and Kannada, belong to the Dravidian

© Bharathi Raja Chakravarthi, Mihael Arcan, and John P. McCrae;
licensed under Creative Commons License CC-BY

languages with scarcely available machine-readable resources. We consider these languages as under-resourced in the context of machine translation (MT) for our research.

Due to the lack of parallel corpora, MT systems for under-resourced languages are less studied. In this work, we attempt to investigate the approach of Multilingual Neural Machine Translation (NMT) [16], in particular, the *multi-way* translation model [13], where multiple sources and target languages are trained simultaneously. This has been shown to improve the quality of the translation, however, in this work, we focus on languages with different scripts, which limits the application of these multi-way models. In order to overcome this, we investigate if converting them into a single script will enable the system to take advantage of the phonetic similarities between these closely-related languages.

Closely-related languages refer to languages that share similar lexical and structural properties due to sharing a common ancestor [33]. Frequently, languages in contact with other language or closely-related languages like the Dravidian, Indo-Aryan, and Slavic family share words from a common root (*cognates*), which are highly semantically and phonologically similar. Phonetic transcription is a method for writing the language in other script keeping the phonemic units intact. It is extensively used in speech processing research, text-to-speech, and speech database construction. Phonetic transcription into a single script has the advantage of collecting similar words at the phoneme level. In this paper, we study this hypothesis by transforming Dravidian scripts into the Latin script and IPA. We study the effect of different orthography on NMT and show that coarse-grained transcription to Latin script outperforms the more fine-grained IPA and native script on multilingual NMT system. Furthermore, we study the usage of sub-word tokenization [38], which has been shown to improve machine translation performance. In combination with sub-word tokenization, phonetic transcription of parallel corpus shows improvement over the native script experiments.

Our proposed methodology allows the creation of MT systems from under-resourced languages to English and in other direction. Our results, presented in Section 5, show that phonetic transcription of parallel corpora increases the MT performance in terms of the BLEU [31], METEOR [3] and chrF [32] metric [9]. Multilingual NMT with closely-related languages improve the score and we demonstrate that transliteration to Latin script outperforms the more fine-grained IPA.

2 Related work

As early as [4], researchers have looked into translation between closely-related languages such as from Czech-Russian RUSLAN and Czech-Slovak CESILKO [17] using syntactic rules and lexicons. The closeness of the related languages makes it possible to obtain a better translation by means of simpler methods. But both systems were rule-based approaches and bottlenecks included complexities associated with using a word-for-word dictionary translation approach. Nakov and Ng [30] proposed a method to use resource-rich closely-related languages to improve the statistical machine translation of under-resourced languages by merging parallel corpora and combining phrase tables. The authors developed a transliteration system trained on automatically-extracted likely cognates for Portuguese into Spanish using systematic spelling variation.

Popović et al. [34] created an MT system between closely-related languages for the Slavic language family. Language-related issues between Croatian, Serbian and Slovenian are explained by [33]. Serbian is digraphic (uses both Cyrillic and Latin Script), the other two are written using only the Latin script. For the Serbian language transliteration without

loss of information is possible from Latin to Cyrillic script because there is a one-to-one correspondence between the characters. The statistical phrase-based SMT system, Moses [23], was used for MT training in these works. In contrast, the Dravidian languages in our study do not have a one-to-one correspondence with the Latin script.

Previous proposed works on NMT, specifically on low-resource [41, 10] or zero-resource MT [20, 15], experimented on languages which have large parallel corpora. These methods used third languages as pivots and showed that translation quality is significantly improved. Although the results were promising, the success of NMT depends on the quality and scale of available parallel corpora from the pivot or third language. The third or pivot language of choice in previous works were well-resourced languages like English, German, French but many under-resourced languages have very different syntax and semantic structure to these languages. We use languages belonging to the same family which shares many linguistic features and properties to mitigate this problem. In previous works, the languages under study shared the same or similar alphabets but, in our research, we deal with the languages which have entirely different orthography.

Machine transliteration [22] is a common method for dealing with names and technical terms while translating into another language. Some languages have special phonetic alphabets for writing foreign words or loanwords. Cherry and Suzuki [11] use transliteration as a method to handle out-of-vocabulary (OOV) problems. To remove the script barrier, Bhat et al. [7] created machine transliteration models for the common orthographic representation of Hindi and Urdu text. The authors have transliterated text in both directions between Devanagari script (used to write the Hindi language) and Perso-Arabic script (used to write the Urdu language). The authors have demonstrated that a dependency parser trained on augmented resources performs better than individual resources. The authors have shown that there was a significant improvement in BLEU (Bilingual Evaluation Understudy) score and shown that the problem of data sparsity is reduced. In the work by [8], the authors translated lexicon induction for a heavily code-switched text of historically unwritten colloquial words via loanwords using expert knowledge with just language information. Their method is to take word pronunciation (IPA) from a donor language and convert them in the borrowing language. This shows improvements in BLEU score for induction of Moroccan Darija-English translation lexicon bridging via French loan words.

Recent work by Kunchukuttan et al. [27] has explored orthographic similarity for transliteration. In their work, they have used related languages which shares similar writing systems and phonetic properties such as Indo-Aryan languages. They have shown that multilingual transliteration leveraging similar orthography outperforms bilingual transliteration in different scenarios. Note that their model cannot generate translations; it can only create transliterations. In this work, we focus on multilingual translation of languages which uses different scripts. Our work studies the effect of different orthographies to common script with multilingual NMT.

3 Dravidian languages

Dravidian languages [25] are spoken in the south of India by 215 million people. To improve access to and production of information for monolingual speakers of Dravidian languages, it is necessary to have an MT system from and to English. However, Dravidian languages are under-resourced languages and thus lack the parallel corpus needed to train an NMT system. For our study, we perform experiments on Tamil (ISO 639-1: ta), Telugu (ISO 639-1: te) and Kannada (ISO 639-1: kn). The targeted languages for this work differ in several ways,

although they have nearly the same number of consonants and vowels, their orthographies differ due to historical reasons and whether they adopted the Sanskrit tradition or not [5].

The Tamil script evolved from the Brahmi script, Vatteluttu alphabet, and Chola-Pallava script. It has 12 vowels, 18 consonants, and 1 *aytam* (voiceless velar fricative). The Telugu script is also a descendant of the Southern Brahmi script and has 16 vowels, 3 vowel modifiers, and 41 consonants. The Kannada script has 14 vowels, 34 consonants, and 2 *yogavahakas* (part-vowel, part-consonant). The Kannada and Telugu scripts are most similar, and often considered as a regional variant. The Kannada script is used to write other under-resourced languages like Tulu, Konkani, and Sankethi. Since Telugu and Kannada are influenced by Sanskrit grammar, the number of characters is higher than in the Tamil language. In contrast to Tamil, Kannada, and Telugu inherits some of the affixes from Sanskrit [40, 36, 25]. Each of these has been assigned a unique block in Unicode, and thus from an MT perspective are completely distinct.

4 Experimental Settings

4.1 Data

To train an NMT system for English-Tamil, English-Telugu, and English-Kannada language pairs, we use parallel corpora from the OPUS[1] web-page [39]. OPUS includes large number of translations from the EU, open source projects, the Web, religious texts and other resources. OPUS also contains translations of technical documentation from the KDE, GNOME, and Ubuntu projects. We took the English-Tamil parallel corpora created with the help of Mechanical Turk for Wikipedia documents [35], EnTam corpus [37] and furthermore manually aligned the well-known Tamil text Tirukkural, which contains 2660 lines. Most multilingual corpora come from the parliament debates and legislation of the EU or multilingual countries, but most non-EU languages lack such resources. For our experiments, we combined all the corpus to form a **complete corpus** and split the corpora into an evaluation set containing 1,000 sentences, a validation set containing 1,000 sentences, and a training set containing the remaining sentences shown in Table 1. Following Ha et al. [16], we indicate the language by prepending two tokens to indicate the desired source and target language.

An example of a sentence in English to be translated into Tamil would be:

```
<en> <ta> Translate into Tamil
```

Table 1 Corpus statistics of the **complete corpus** (Collected from OPUS on August 2017) used for MT. (Tokens-En: Total number of tokens in the English side of parallel corpora. Tokens-Dr: Total number of tokens in the Dravidian language side of parallel corpora.)

	Number of sentences	Tokens-English	Tokens-Dravidian
English-Tamil	2,248,685	44,139,295	34,111,290
English-Telugu	224,940	1,386,861	1,714,860
English-Kannada	69,715	504,098	687,413
Total	2,543,340	46,030,254	36,513,563

[1] http://opus.nlpl.eu/

Table 2 Corpus Statistics of the **multi-parallel corpus** used for MT. (Tokens-En: Total number of tokens in the English side of parallel corpora. Tokens-Dr: Total number of tokens in the Dravidian language side of parallel corpora.)

	Number of sentences	Tokens-English	Tokens-Dravidian
English-Tamil	38,930	238,654	153,087
English-Telugu	38,930	238,654	164,335
English-Kannada	38,930	238,654	183,636
Total	116,790	715,962	501,058

Table 3 Orthographic representation of word *blue* in Tamil, Telugu and Kannada shown in native script, Latin script and IPA.

ISO 639-1	Script	Spelling	Transliteration	IPA	English
ka	Kannada	ನೀಲಿ	nili	nili	Blue
ta	Tamil	நீலம்	nilam	n̪i:lam	Blue
te	Telugu	నీలం	nilam	ni:ləm	Blue

4.2 Multi-parallel Corpus

In order to enable the training of the multi-way model, we developed a **multi-parallel corpus**, which consists of only the sentences that are available in all four languages. In this small subset of the complete corpus, most of the sentences for the Dravidian languages came from the translations of technical documents. The English sentences from the bilingual parallel corpora of three languages are aligned by collecting common English sentences from all three languages and their translation in the Dravidian languages. For the one-to-many multilingual models and many-to-one models [14], the parallel corpora were combined to form an English-to-Dravidian (Tamil, Telugu, and Kannada) NMT and Dravidian (Tamil, Telugu, and Kannada)-to-English NMT.

The corpus consists of 38,930 sentences, shown in Table 1. Combined, the corpus used to train multilingual NMT models consists of 116,790 sentences, 715,962 sources (English) tokens, and 501,058 target tokens.

4.3 Transliteration

In this section, we study the hypothesis of transliterating Dravidian scripts into the Latin script. Transliteration is a common method for dealing with technical terms and names while translating into another language. It is an approach where a word in one script is transformed into a different character set while attempting to maintain phonetic correspondence. As most of the Indian languages use different scripts, to take advantage of multilingual NMT models, we converted the Tamil, Telugu and Kannada script into the Latin script for a common representation before merging them into a multilingual corpus. We have used the Indic-trans library[2] [6] to transliterate the Dravidian side of the parallel corpus for three Dravidian languages, namely Tamil, Telugu, and Kannada, into the Latin script. The indic-trans lib produces 92.53 % accuracy for Tamil-English, 92.27 % accuracy for Telugu-English, and 91.89 % accuracy for Kannada-English.

[2] `https://github.com/libindic/indic-trans`

4.4 International Phonetic Alphabet - IPA

The International Phonetic Alphabet (IPA) [19] contains symbols for vowels, consonants and prosodic features, such stress and it is intended to be an accurate phonetic representation for all languages. We use IPA for the phonetic transcription of Dravidian languages into a single representation. We use the Epitran library [29], which is a grapheme-to-phoneme transducer supporting 61 languages. It takes the words as input and provides phonetic transcription in IPA. It has support for Tamil and Telugu but not for Kannada. Therefore, we used the Txt2ipa[3] library for Kannada, which uses a dictionary mapping to convert the Kannada script into IPA script. Table 3 shows the English word *blue* in native script, transliteration and IPA. From the figure, it is clear that the transliteration has more common sub-word units than IPA.

4.5 Translation experiments

We performed our experiments with OpenNMT [21] a toolkit for neural machine translation and neural sequence modeling. After tokenization, we fed the parallel corpora to the OpenNMT preprocessing tools i.e. OpenNMT tokenizer. Preprocessed files were then used to train the models. We used the OpenNMT parameters based on the paper [16] for training, i.e., 4 layers, 1000 for RNN size, bidirectional RNN, and 600-word embedding size, input feeding enabled, batch size of 64, 0.3 dropout probability and a dynamic learning rate decay.

The approach of [16] allows us to integrate the multilingual setting with a single encoder-decoder approach and without modification of the original OpenNMT model. This unified approach to extend the original NMT to multilingual NMT does not require any special treatment of the network during training. We compare the multilingual NMT model with bilingual models for both multilingual corpora and multiway multilingual corpora. Different evaluation sets were used for test multi-way multilingual and multilingual systems.

Table 4 Cosine similarity of the transliteration of the languages under study at character level using the **complete corpus**.

	Latin script	**IPA**
Tamil-Telugu	0.9790	0.7166
Tamil-Kannada	0.9822	0.5827
Telugu-Kannada	0.9846	0.8588

Table 5 Cosine similarity of the transliteration of the languages under study at character level using the **multi-parallel corpus**.

	Latin script	**IPA**
Tamil-Telugu	0.9867	0.6769
Tamil-Kannada	0.9825	0.5602
Telugu-Kannada	0.9855	0.5679

[3] `https://github.com/arulalant/txt2ipa`

Table 6 BLEU (B), METEOR (M) and chrF (C) scores are illustrated for systems trained with native script, Latin script and IPA. Native script is different for Tamil, Telugu and Kannada. Latin script and IPA are common script representations. Best results for each systems are shown in bold.

	Native Script			Latin Script			IPA		
	B	M	C	B	M	C	B	M	C
Bilingual systems results trained at word level									
En-Ta	**40.32**	**34.79**	**62.70**	39.7	23.48	50.10	30.67	26.37	45.27
En-Te	20.15	21.37	40.93	**20.43**	**21.42**	**41.20**	19.3	20.06	40.09
En-Kn	**28.15**	**33.53**	**60.20**	28.13	23.46	42.96	27.11	33.50	50.78
Ta-En	**32.21**	**25.65**	**44.68**	30.72	24.78	43.60	31.2	25.29	43.60
Te-En	**16.24**	**28.36**	33.22	17.96	11.84	31.26	12.65	29.23	**44.01**
Kn-En	**25.93**	**22.20**	**41.88**	23.89	20.81	39.82	20.52	18.65	17.02
Multilingual systems results trained at word level									
En-Ta	43.6	34.57	64.58	**44.23**	**35.48**	**65.02**	32.94	23.86	47.03
En-Te	23.69	23.37	42.32	**23.98**	23.93	**42.49**	22.35	**25.98**	42.86
En-Kn	28.82	33.62	**62.73**	**31.71**	35.03	46.12	30.59	**36.45**	53.94
Ta-En	29.8	24.83	46.64	**35.66**	**28.43**	**47.44**	33.86	27.34	46.89
Te-En	17.82	32.34	**56.61**	**22.95**	**24.68**	36.14	16.39	24.34	48.29
Kn-En	25.11	18.50	42.60	**28.31**	**27.63**	**42.95**	24.46	24.54	19.83

5 Results

5.1 Comparison of transliteration methodologies

While it is clear that IPA is generally a more fine-grained transliteration than the transliteration to Latin script, we wished to quantitatively evaluate this difference. Thus, we took the complete corpus for each language and for each character (Unicode codepoint) that occured in the texts, we calculated its total frequency c_f^l. We then calculated the cosine similarity between the two languages, l_1, l_2, e.g.,

$$sim^{l_1,l_2} = \frac{\sum_c f_c^{l_1} f_c^{l_2}}{\sqrt{\sum_c (f_c^{l_1})^2 \sum_c (f_c^{l_2})^2}}$$

Table 4 and 5 shows the statistics of the cosine similarity at the character level, showing that our intuition that the Latin transliteration is much more coarse-grained is well-founded as the results show that the Latin script produces a cosine similarity of about 0.98 for these three languages whereby the IPA score is lower compared to the Latin script.

To further validate this, we show in Table 3 the word *blue* in all the three languages. The root word *nil* is the same in all the languages whereby Tamil and Telugu have commonality at the whole word level. It is clear that there are far fewer commonalities in the IPA transliteration than in the Latin script transliteration.

5.2 Translation Results

Using the data, settings, and metrics described above, we investigated the impact of phonetic transcription on the machine translation of closely-related languages in multilingual NMT. We trained 54 bilingual and 18 multilingual systems corresponding to training policies and languages discussed above. All the systems were trained for 13 epochs. We use BLEU [31], METEOR [3] and chrF [32] metrics for the translation evaluation. BLEU is an

Table 7 BLEU (B), METEOR (M) and chrF (C) scores are shown for systems trained with native script, Latin script and IPA for **multi-parallel corpora with different evaluation set**. Native script is different for Tamil, Telugu and Kannada. Latin script and IPA are common script representations. Best results for each systems are shown in bold.

	Native Script			**Latin Script**			**IPA**		
	B	M	C	B	M	C	B	M	C
Bilingual systems results trained at word level									
En-Ta	31.91	22.94	43.77	**36.18**	**31.24**	**49.45**	28.67	22.92	32.35
En-Te	37.70	36.53	45.39	**38.67**	**34.12**	**48.44**	30.39	32.21	38.35
En-Kn	25.45	12.67	38.49	**26.51**	**28.66**	**39.87**	23.37	16.55	35.66
Ta-En	31.49	**37.61**	41.33	**34.75**	37.15	**43.24**	36.61	36.24	37.59
Te-En	35.30	32.23	49.35	36.44	34.69	42.72	**38.84**	**37.65**	**49.40**
Kn-En	**33.14**	21.71	44.76	30.17	**32.08**	**51.71**	24.87	18.63	45.53
Multilingual system results trained at word level									
En-Ta	37.32	38.94	50.56	**41.99**	**43.67**	49.11	38.45	39.66	**52.38**
En-Te	38.75	38.66	52.83	**39.67**	**42.75**	**56.44**	32.39	32.21	43.35
En-Kn	35.67	28.03	55.12	**37.85**	**32.43**	**60.53**	34.93	26.22	57.38
Ta-En	**36.03**	**32.32**	**54.46**	34.53	31.33	52.55	30.47	27.74	52.23
Te-En	34.22	31.17	53.14	**42.42**	**33.72**	**56.77**	30.72	25.82	52.28
Kn-En	32.15	46.65	59.49	**36.47**	**33.79**	**63.79**	34.59	41.06	56.12
Bilingual systems results trained at sub-word level tokenization									
En-Ta	36.11	20.30	53.43	**46.82**	**39.55**	**62.13**	43.63	36.36	61.90
En-Te	37.53	36.24	44.56	**39.47**	**36.34**	**58.45**	38.2	33.76	69.06
En-Kn	35.99	27.71	**55.37**	**39.20**	**42.94**	52.07	30.77	27.29	53.11
Ta-En	32.56	**23.42**	29.00	**36.62**	23.12	**44.35**	29.75	22.47	23.61
Te-En	36.12	18.93	56.63	**38.82**	**35.01**	**54.39**	39.5	25.95	37.65
Kn-En	34.85	29.26	43.86	**34.98**	**38.92**	**51.65**	33.87	24.27	45.00
Multilingual systems results trained at sub-word level tokenization									
En-Ta	39.25	31.91	**62.18**	**40.77**	**36.66**	56.52	31.34	27.32	52.16
En-Te	37.63	38.16	64.20	**38.33**	**43.34**	**67.45**	35.20	23.76	59.06
En-Kn	37.17	30.31	56.39	**37.85**	**37.08**	**59.03**	53.21	29.93	54.46
Ta-En	**37.18**	**34.69**	**57.58**	35.52	31.27	55.01	36.86	32.78	56.68
Te-En	**35.79**	**23.67**	46.76	29.61	23.28	46.97	28.43	20.39	**37.24**
Kn-En	**34.15**	39.84	62.19	30.53	**40.74**	**64.29**	27.36	24.56	29.38

automatic evaluation technique which is a geometric mean of n-gram precision. It is language-independent, fast, and shows a good correlation with human judgment. It is extensively used for various MT evaluations. The METEOR metric was designed to address the drawbacks of BLEU. We also used the chrF metric to study system output at the character level which uses F-score based on character n-grams. It is absolutely language independent and also tokenization independent.

5.2.1 Analysis of Latin script results

In order to provide a consistent evaluation of results, we wished to compare the system outputs using the native script in all settings, instead of using the output translations in IPA and Latin script. Thus, we back-transliterated the generated translations using the Indic-trans library from Latin script to native script and ran the evaluation metrics for

Table 8 Manual evaluation results of 50 sentences for translation between English and Tamil.

	Ideal	Acceptable	Possibly Acceptable	Unacceptable
Native Script				
En-Ta	8	11	14	17
Ta-En	8	13	18	11
Transliteration				
En-Ta	8	14	12	16
Ta-En	9	13	21	7
IPA				
En-Ta	6	14	17	13
Ta-En	3	18	18	11

both the corpora. Table 6 and 7 compare the results of various NMT generated translation in BLEU, METEOR, and chrF. We observe that the translations from Latin script based system provides an improvement in terms of BLEU, METEOR and chrF scores for translation from English to Tamil, Telugu, and Kannada for the bilingual systems for the multi-parallel corpus. This trend continues in the evaluation scores for the multilingual model as well. The multilingual systems outperform the baseline bilingual systems trained on the native script. The results are shown in Table 7. The METEOR and chrF score also show the same trend as the BLEU scores. Compared to the bilingual NMT system based on the native script, the multilingual NMT system based on the Latin script has improvement in the BLEU score for translation from English to Dravidian languages.

In the other direction, i.e., from Tamil, Telugu, and Kannada to English, the results are different. The Tamil→English model, based on the native script, has a higher BLEU score that the Latin Script for the multi-parallel corpus. For the Telugu→English model, based on Latin script, there is an improvement in BLEU score and Kannada-English models based on Latin script there is an improvement in BLEU score. The multilingual model of Tamil-English and Telugu-English have higher BLEU score based on the native script than the Latin script, except for the Kannada-English model where the Latin script based models outperform the native script based models. The might be the cause of translating from many languages to single languages in our case English.

5.2.2 Analysis of IPA results

To back-transcribe IPA translations into the native script, we trained an NMT system using the IPA corpus and native script corpus as a parallel resource; this was to ensure that the comparison is fair between the different transliterations. For the IPA-Tamil (Script) system, we got the 90.24 BLEU-1, and 93.07 chrF scores. BLEU-1 94.11, and chrF 94.37 for IPA-Telugu. For the IPA-Kannada BLEU-1 score was 90.51, and chrF was 89.34. We then transcribed the evaluation data to a native script using the above NMT systems. Despite the promising results in multilingual NMT, IPA results are lower compared to Latin script based systems. We observed that the scores of BLEU, METEOR, and chrF are lower than the results based on the native script in bilingual NMT translations in Table 6 and 7. It is noticeable that the scores from Dravidian languages to English trained with IPA representations did not improve the translation quality. This is due to the fact that the IPA representation was very detailed at the phonetic level than the Latin script transliteration.

5.3 Comparing BPE with word level models

There are two broad approaches to tokenize the corpora for MT. The first approach involves word level tokenization and the second is sub-word level tokenization (Byte Pair Encoding). At sub-word level, closely related languages have a high degree of similarity, thus makes it possible to effectively translate shared sub-words [26]. Byte Pair Encoding (BPE) avoids OOV issues by representing a more frequent sub-word as atomic units [38]. We train our models on space-separated tokens (words) and sub-word units. Sub-word tokenization is proven to improve the results in the translation of rare and unseen words for the language pairs like English→German, English→French and other languages [38]. Our experiments on the generated translations of the models based on the BPE corpus reveals that the systems based on Latin script have higher BLEU score in all targeted translation direction i.e. from English to Dravidian language and vice versa. Moreover, by analyzing the METEOR and chrF scores we note that systems, based on the Latin script using sub-word segmented corpora effectively reduce the translation errors. Again, we observed improvements from English into Dravidian languages but a drop in results for the other direction. Results for the model trained at the sub-word level are shown in Table 7. The transliteration-based multilingual system outperforms both the native and the IPA script based multilingual system. These results indicated that the coarse-grained transliteration to Latin script gives an improvement of MT results by better taking advantage of closely-related languages.

6 Error Analysis

We observed an improved performance of Latin script compared to native script and IPA, which is due to the limited number of characters, which better represents the phonological similarity of these languages. We see that the Latin transliteration mostly outperforms both the native script and the IPA transliteration and furthermore that the sub-word tokenization also improves performance. Surprisingly, the combination of these methodologies does not seem to be effective.

We can explain this by the example of the words ‘nilam’ and ‘nili’, which when we apply sub-word tokenization become ‘nil’ and ‘am’ or ‘i’. While Tamil and Telugu have similar morphology for this word, the common token of ‘am’ and ‘i’ are difficult to map to Kannada.

For word-level representation in native script, the number of translation units can increase with corpus size, especially for morphologically rich languages, like Dravidian languages which lead to many OOVs, and thus, a single script with sub-word units addresses the data sparsity issue most effectively.

We performed a manual analysis of the outputs generated by the different systems. Table 8 show the results of manual evaluation. We used four categories based on the work by [12]:

Ideal. Grammatically correct with all information accurately transferred.
Acceptable. Comprehensible with the accurate transfer of all important information.
Possibly Acceptable. Some information transferred accurately.
Unacceptable. Not comprehensible and/or not much information transferred accurately.

From the manual analysis, we found out that the native script and transliteration methods are more similar in terms of ideal and acceptable translation, while IPA has fewer ideal results due to errors at the character level. The unacceptable case is high in results from native script translation due to many out of vocabulary terms. All three methods have similar numbers of acceptable and possibly acceptable cases.

7 Conclusion

In this work, we described our experiments on translation across different orthographies for under-resourced languages such as Tamil, Telugu, and Kannada. We show that in the Tamil, Telugu, and Kannada to English translation direction the translation quality of bilingual NMT and multilingual NMT systems improves. In order to remove the orthographic differences between languages in the same family, we performed transcription from a native script into Latin script and IPA. We demonstrated that the phonetic transcription of parallel corpora of closely-related languages shows better results and that the multilingual NMT with phonetic transcription to Latin script performs better than IPA transliteration. This can be explained due to the coarse-grained natures of the transliteration, which produce more similarity at the character level in the target languages, which we proved by evaluating the cosine similarity of the character frequencies.

References

1 Steven Abney and Steven Bird. The Human Language Project: Building a Universal Corpus of the World's Languages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 88–97. Association for Computational Linguistics, 2010. URL: http://www.aclweb.org/anthology/P10-1010.

2 Iñaki Alegria, Xabier Artola, Arantza Diaz De Ilarraza, and Kepa Sarasola. Strategies to develop Language Technologies for Less-Resourced Languages based on the case of Basque, 2011.

3 Satanjeev Banerjee and Alon Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72. Association for Computational Linguistics, 2005. URL: http://www.aclweb.org/anthology/W05-0909.

4 Alevtina Bemova, Karel Oliva, and Jarmila Panevova. Some Problems of Machine Translation Between Closely Related Languages. In *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*, 1988. URL: http://www.aclweb.org/anthology/C88-1010.

5 Kamadev Bhanuprasad and Mats Svenson. Errgrams - A Way to Improving ASR for Highly Inflected Dravidian Languages. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*, 2008. URL: http://www.aclweb.org/anthology/I08-2113.

6 Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. IIIT-H System Submission for FIRE2014 Shared Task on Transliterated Search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA, 2015. ACM. doi:10.1145/2824864.2824872.

7 Riyaz Ahmad Bhat, Irshad Ahmad Bhat, Naman Jain, and Dipti Misra Sharma. A House United: Bridging the Script and Lexical Barrier between Hindi and Urdu. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 397–408, 2016. URL: http://aclweb.org/anthology/C/C16/C16-1039.pdf.

8 Michael Bloodgood and Benjamin Strauss. Acquisition of Translation Lexicons for Historically Unwritten Languages via Bridging Loanwords. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 21–25. Association for Computational Linguistics, 2017. doi:10.18653/v1/W17-2504.

9 Bharathi Raja Chakravarthi, Mihael Arcan, and John P. McCrae. Improving Wordnets for Under-Resourced Languages Using Machine Translation. In *Proceedings of the 9th Global WordNet Conference*. The Global WordNet Conference 2018 Committee, 2018. URL: http://compling.hss.ntu.edu.sg/events/2018-gwc/pdfs/GWC2018_paper_16.

10 Yun Chen, Yang Liu, Yong Cheng, and Victor O.K. Li. A Teacher-Student Framework for Zero-Resource Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1925–1935. Association for Computational Linguistics, 2017. doi:10.18653/v1/P17-1176.

11 Colin Cherry and Hisami Suzuki. Discriminative Substring Decoding for Transliteration. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1066–1075. Association for Computational Linguistics, 2009. URL: http://www.aclweb.org/anthology/D09-1111.

12 Deborah Coughlin. Correlating automated and human assessments of machine translation quality. In *In Proceedings of MT Summit IX*, pages 63–70, 2003.

13 Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875. Association for Computational Linguistics, 2016. doi:10.18653/v1/N16-1101.

14 Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T. Yarman Vural, and Yoshua Bengio. Multi-way, Multilingual Neural Machine Translation. *Comput. Speech Lang.*, 45(C):236–252, September 2017. doi:10.1016/j.csl.2016.10.006.

15 Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. Zero-Resource Translation with Multi-Lingual Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277. Association for Computational Linguistics, 2016. doi:10.18653/v1/D16-1026.

16 Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. In *Proceedings of the International Workshop on Spoken Language Translation*, 2016. URL: http://workshop2016.iwslt.org/downloads/IWSLT_2016_paper_5.pdf.

17 Jan Hajic, Jan Hric, and Kubon Vladislav. Machine Translation of Very Close Languages. In *Sixth Applied Natural Language Processing Conference*, 2000. URL: http://www.aclweb.org/anthology/A00-1002.

18 Auður Hauksdóttir. An Innovative World Language Centre : Challenges for the Use of Language Technology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA), 2014. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/795_Paper.pdf.

19 International Phonetic Association. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

20 Johnson, Melvin and Schuster, Mike and Le, Quoc V. and Krikun, Maxim and Wu, Yonghui and Chen, Zhifeng and Thorat, Nikhil and Viégas, Fernanda and Wattenberg, Martin and Corrado, Greg and Hughes, Macduff and Dean, Jeffrey. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. URL: http://aclweb.org/anthology/Q17-1024.

21 Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics, 2017. URL: http://www.aclweb.org/anthology/P17-4012.

22 Kevin Knight and Jonathan Graehl. Machine Transliteration. *Computational Linguistics*, 24(4), 1998. URL: http://www.aclweb.org/anthology/J98-4003.

23 Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics, 2007. URL: http://www.aclweb.org/anthology/P07-2045.

24 Steven Krauwer. The basic language resource kit (BLARK) as the first milestone for the language resources roadmap. *Proceedings of SPECOM 2003*, pages 8–15, 2003.

25 Arun Kumar, Ryan Cotterell, Lluís Padró, and Antoni Oliver. Morphological Analysis of the Dravidian Language Family. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 217–222. Association for Computational Linguistics, 2017. URL: http://aclweb.org/anthology/E17-2035.

26 Anoop Kunchukuttan and Pushpak Bhattacharyya. Learning variable length units for SMT between related languages via Byte Pair Encoding. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 14–24. Association for Computational Linguistics, 2017. URL: http://aclweb.org/anthology/W17-4102.

27 Anoop Kunchukuttan, Mitesh Khapra, Gurneet Singh, and Pushpak Bhattacharyya. Leveraging Orthographic Similarity for Multilingual Neural Transliteration. *Transactions of the Association for Computational Linguistics*, 6:303–316, 2018. URL: http://aclweb.org/anthology/Q18-1022.

28 Mike Maxwell and Baden Hughes. Frontiers in Linguistic Annotation for Lower-Density Languages. In *Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006*, pages 29–37. Association for Computational Linguistics, 2006. URL: http://www.aclweb.org/anthology/W06-0605.

29 David R. Mortensen, Siddharth Dalmia, and Patrick Littell. Epitran: Precision G2P for Many Languages. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, May 2018. European Language Resources Association (ELRA).

30 Preslav Nakov and Hwee Tou Ng. Improved Statistical Machine Translation for Resource-Poor Languages Using Related Resource-Rich Languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1367. Association for Computational Linguistics, 2009. URL: http://www.aclweb.org/anthology/D09-1141.

31 Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002. URL: http://www.aclweb.org/anthology/P02-1040.

32 Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395. Association for Computational Linguistics, 2015. doi:10.18653/v1/W15-3049.

33 Maja Popović, Mihael Arcan, and Filip Klubička. Language Related Issues for Machine Translation between Closely Related South Slavic Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 43–52. The COLING 2016 Organizing Committee, 2016. URL: http://www.aclweb.org/anthology/W16-4806.

34 Maja Popović and Nikola Ljubešić. Exploring cross-language statistical machine translation for closely related South Slavic languages. In *Proceedings of the EMNLP'2014 Workshop on Language Technology for Closely Related Languages and Language Variants*, pages 76–84. Association for Computational Linguistics, 2014. doi:10.3115/v1/W14-4210.

35 Matt Post, Chris Callison-Burch, and Miles Osborne. Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 401–409. Association for Computational Linguistics, 2012.

36 P. Prakash and R. Malatesha Joshi. *Orthography and Reading in Kannada: A Dravidian Language*, pages 95–108. Springer Netherlands, Dordrecht, 1995. doi:10.1007/978-94-011-1162-1_7.

37 Loganathan Ramasamy, Ondřej Bojar, and Zdeněk Žabokrtský. Morphological Processing for English-Tamil Statistical Machine Translation. In *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012)*, pages 113–122, 2012.

38 Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics, 2016. doi:10.18653/v1/P16-1162.

39 Jorg Tiedemann and Lars Nygaard. The OPUS Corpus - Parallel and Free: http://logos.uio.no/opus . In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. European Language Resources Association (ELRA), 2004. URL: http://www.lrec-conf.org/proceedings/lrec2004/pdf/320.pdf.

40 Devadath V V and Dipti Misra Sharma. Significance of an Accurate Sandhi-Splitter in Shallow Parsing of Dravidian Languages. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 37–42. Association for Computational Linguistics, 2016. doi:10.18653/v1/P16-3006.

41 Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575. Association for Computational Linguistics, 2016. doi:10.18653/v1/D16-1163.

CoNLL-Merge: Efficient Harmonization of Concurrent Tokenization and Textual Variation

Christian Chiarcos
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
http://www.acoli.informatik.uni-frankfurt.de/
chiarcos@informatik.uni-frankfurt.de

Niko Schenk
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
schenk@informatik.uni-frankfurt.de

Abstract

The proper detection of tokens in of running text represents the initial processing step in modular NLP pipelines. But strategies for defining these minimal units can differ, and conflicting analyses of the *same* text seriously limit the integration of subsequent linguistic annotations into a shared representation. As a solution, we introduce *CoNLL Merge*, a practical tool for harmonizing TSV-related data models, as they occur, e.g., in multi-layer corpora with non-sequential, concurrent tokenizations, but also in ensemble combinations in Natural Language Processing. *CoNLL Merge* works unsupervised, requires no manual intervention or external data sources, and comes with a flexible API for fully automated merging routines, validity and sanity checks. Users can chose from several merging strategies, and either preserve a reference tokenization (with possible losses of annotation granularity), create a common tokenization layer consisting of minimal shared subtokens (loss-less in terms of annotation granularity, destructive against a reference tokenization), or present tokenization clashes (loss-less and non-destructive, but introducing empty tokens as place-holders for unaligned elements). We demonstrate the applicability of the tool on two use cases from natural language processing and computational philology.

2012 ACM Subject Classification Applied computing → Format and notation; Applied computing → Document management and text processing; Applied computing → Annotation; Software and its engineering → Interoperability

Keywords and phrases data heterogeneity, tokenization, tab-separated values (TSV) format, linguistic annotation, merging

Digital Object Identifier 10.4230/OASIcs.LDK.2019.7

Supplement Material https://github.com/acoli-repo/conll

Funding The research described in this paper was supported by the project *Linked Open Dictionaries* (LiODi, 2015-2020), funded by the German Ministry for Education and Research (BMBF) and the project *Machine Translation and Automated Analysis of Cuneiform Languages* which is generously funded by the German Research Foundation (DFG), the Canadian Social Sciences and Humanities Research Council, and the National Endowment for the Humanities through the T-AP Digging into Data Challenge.

1 Motivation

Linguistic annotations of running text exhibit a great diversity and comprise, among others, part-of-speech tags, phrasal chunks, syntactic parses, semantic roles, or discourse relations. *Tokenization* as the initial pre-processing step is concerned with the proper detection and segmentation of application-specific, minimal units, i.e. *tokens*, and represents the basis for subsequent annotations. Tokens can be typified by words (lexemes or morphemes) or other methodologically-informed symbols (numbers, alpha-numerics and punctuation), and


© Christian Chiarcos and Niko Schenk;
licensed under Creative Commons License CC-BY

have various properties: In many applications, they constitute the basis for "word" distance measurements, e.g., Normalized Levenshtein [29] and related similarity tasks involving word embeddings [17]. Beyond that, in many annotation tools and their corresponding formats, the order of tokens provides a timeline for the sequential order of structural elements [18].

Similarly, multi-layer formats refer to tokens in order to define the absolute position of annotation elements, and only by reference to a single privileged token layer (or an alternative base segmentation), annotations from different layers can be put into relation with each other [3]. On the single privileged token layer, tokens are organized in a total order, they cover the full annotated portion of the primary data, and represent the smallest unit of annotation. This aspect is especially important for the study of richly annotated syntactic and semantic resources, as an integration and serialization of linguistic annotations produced by *different* tools is usually established by reference to the token layer. Unfortunately, different annotation routines on the *same* texts oftentimes rely on *concurrent* tokenization schemes, which crucially requires efforts for harmonization. This is of particular relevance for NLP tools which need to draw on multiple linguistic annotations but for which concurrent information (potentially stored in alignment-incompatible, distinct data formats) heavily complicate their development process.

Our Contribution. Based on these observations, we argue that with the availability of robust conversion and flexible merging routines for standardized CoNLL and TSV-related data models, complex NLP tools that rely on a multitude of linguistic annotations can be realized in a more straightforward way. To this end, we introduce *CoNLL Merge*, a fully automated, application-independent merging routine for linguistic annotations based on different underlying tokenizations of the same text. The theoretical basis for our approach is described in [22], however, while [22] build their implementation of a highly complex standoff XML formalism with limited use in natural language processing and the language sciences, our implementation focuses on one-word-per-line (OWPL) tab separated value (TSV) formats, a simple formalism with wide application in corpus linguistics, lexicography and natural language processing, most famously associated with the long series of CoNLL shared tasks.

We are aware that project- or application-specific solutions for automated tokenization harmonization do exist (cf. Sect. 5). In opposition to these, *CoNLL Merge* provides a generic solution which does not only allows to merge files in *any* OWPL TSV format without manual interference, but which also allows to define the merging strategy – depending on whether the user prefers reversibility or keeping/enforcing a particular tokenization. We illustrate the practicability of our approach on a collection of annotated texts from the Wall Street Journal-based corpora that are in the intersection of several corpora with independent manual annotations, frequently with concurrent tokenizations. A second experiment on historical texts demonstrates the robustness of CoNLL Merge against textual variation *beyond* tokenization mismatches.

The paper is structured as follows: Section 2 describes alignment strategies for plain (tokenized) text, Section 3 describes the merging of the associated annotations, and Section 4 provides an evaluation.

2 Aligning Tokenizations

Among both efforts to manually create annotations for linguistics and philology and NLP tools to automatize such annotations, we find a remarkable band-width and variation even within a single language. If multiple annotators (manual or automated) are applied the

same piece of text, they choose (or require) a specific tokenization strategy – and this may deviate greatly from the tokenization adopted by another. Tokenization strategies differ with respect to the research question or application of interest (e.g., tagging, parsing, information extraction), and can be divided into morphosyntactic, full syntactic, and morphology-based analyses. For instance, tokenizations can drastically disagree as in the examples to the right for the text *the attorney general's office* [6]:

1. [attorney] [general's] British National Corpus [2]
2. [attorney] [general][']s] Tnt Tagger [1]
3. [attorney] [general]['s] Penn TreeBank [14]
4. [attorney general]['s] Protein Name Tagger [28]

Crucially, when dealing with multiple linguistic annotations on top of concurrent tokenizations of the *same* text, efforts for harmonization are required. Here, we focus on strategies for their automated alignment. The handling of associated annotations is subject to Sect. 3.

2.1 Identity-Based Alignment

The primary strategy for aligning concurrent tokenizations is based on string identity between different variants of the same text: Even if token boundaries have been inserted and whitespaces have been normalized, we can normally assume that textual content remains untouched.[1]

For string alignment, we build on existing diff implementations, most notably Myer's Diff [19, 15].[2] The scope of our implementation differs from standard Un*x functionalities in that the basis of comparison is the token rather than the line. In default alignment, tokenization mismatches are described by insertions and deletions of tokens. Thus, concerning the alignment between two files, `FILE1` and `FILE2`, three cases have to be distinguished, which we handle as follows:

1. 1:1 alignment
2. 1:0 alignment: For $n:0$ alignment, spell out n lines (tokens) with 1:0 alignments.
3. 0:1 alignment: For $0:m$ alignment, spell out m lines (tokens) with 0:1 alignments.

An $n:m$ alignment will thus be represented by a sequence of $n:0$ (1:0) and $0:m$ (0:1) alignments. In addition to this default merging, we support two merging strategies based on string identity:

forced. Enforce a 1:1 (or 1:0) alignment by concatenating the last 1:1-aligned token from `FILE2` (and its annotations) with those of the following 0:m alignments.

split. Enforce a dense alignment based on maximal common substrings: After default alignment, define an alignment window as a sequence of tokens that start with 1:1 alignment, followed by a sequence of 0:1 and 1:0 alignments. Within that window, perform a character-level (rather than a token-level) diff and aggregate consecutive sequences of 1:1 character alignments into maximal common substrings.

[1] In fact, this is often not true, as annotation tools may replace reserved characters with special symbols or escape sequences, enforce different character encodings or drop, for example, diacritics. However, our implementation has been proven robust against such changes.

[2] At the moment, we employ the implementation from Java *diff utils* by Dmitry Naumenko, `https://github.com/dnaumenko/java-diff-utils`.

It should be noted that the split strategy does not guarantee to arrive at 1:1 alignments in case of character insertions or deletions.[3] In those cases, another force alignment can be applied to eliminate $0:1$ alignments. Likewise, split alignment can be applied after force alignment to reduce the number of $1:0$ alignments. In either case, another challenge is the treatment of the associated annotations.

2.2 Similarity-Based Alignment

CoNLL-Merge was originally intended to cope with conflicting annotations of the same text. However, initial experiments showed that it is also directly applicable as a collator, i.e., a tool that identifies corresponding and deviating text passages and merges them into a common representation. Collators are frequently used in various branches of computational philology, e.g., to identify patterns of re-use and adaptation among different textual fragments (intertextual relations) or manuscript and edition genealogy (stemmatology). Taking CollateX (see Sect. 4.2 below) as an example, it resembles CoNLL Merge in building on existing diff implementations, it exceeds plain diff functionalities in providing convenient user interfaces and visualizations. Unlike CoNLL Merge, CollateX is restricted to plain text and does not provide a way for harmonizing annotations.

Initial experiments for applying CoNLL Merge to philological data have been performed against a small collection of medieval manuscripts written in different orthographies in Middle Low German (cf. Sect. 4.2 below). CoNLL Merge successfully achieved an alignment despite the fact that these texts deviated in their choice of words and in editorial changes such as insertions and deletions of large portions of texts. However, the method was obviously not sufficiently robust against deviating orthographies (a common problem in medieval texts). In order to improve its usability in Digital Humanities contexts, we provide an additional merging strategy based on string similarity rather than identity, based on minimal edit distance, resp., Levenshtein distance [29]. Like force and split, Levenshtein alignment is applied after default alignment was applied to determine alignment windows (non-aligned tokens preceded and/or followed by identity-aligned tokens).

levenshtein. Within the alignment window, determine the source and target token pair with minimum Levenshtein distance. Accept this as alignment and create novel alignment windows before and after the aligned words. Iterate until no more $n:m$ alignments (i.e., sequences of $n:0$ and $0:m$ alignments) remain.

As our application of Levenshtein alignment does not support crossing edges, it does often not produce a 1:1 alignment.

3 Merging Annotations

For merging annotations in one-word-per-line formats, we focus on tabular formats using tab-separated values (TSV), as widely used in corpus linguistics and lexicography, but also in natural language processing (most notably in the context of the long series of CoNLL Shared Tasks).

[3] For identical text, apparent insertions or deletions can arise from different escaping strategies, e.g., the replacement of double quotes with two single quotes, or encoding differences, e.g., the direct encoding of UTF-8 characters or their representation as XML entities.

3.1 The CoNLL Format Family

Since 1999, the Conference on Natural Language Learning[4] (CoNLL) has established a tradition of annual shared tasks in which training and test data is provided by the organizers, thereby facilitating the systematic evaluation of participating tools.[5] With their continuous progression in terms of linguistic complexity, the shared tasks reflect the maturation of statistical NLP, the dominating paradigm of computational linguistics in the 2000s. In many cases, successful participants established reference tools, and – as it allowed for comparative evaluation – the underlying, standardized formats continued to be supported by succeeding NLP tools, which in fact has reinforced the global importance of the CoNLL format family within the language processing community and which thus represents the core basis for the merging routine described in this paper.

3.2 CoNLL Merge

We offer a lightweight Java package for sanity checks, format testing, producing, manipulating and – most notably – merging of TSV files. On the one hand, `CoNLLChecks` can be applied for selected validity checks on a set of CoNLL files.[6] This is particularly useful as a preprocessing step before the actual merging routine. On the other hand, for merging on token and subtoken level itself, most importantly, `CoNLLAlign` establishes the core interface to the implementation. It takes two files to be merged as input (`FILE1 FILE2`) allowing for the following options:

- default
  1:1 alignment: write the `FILE1` line, write a tabulartor, write the `FILE2` line.
  1:0 alignment: write the `FILE1` line, fill up missing `FILE2` columns with the placeholder (`?`).
  0:1 alignment: create an "empty" token (∗`RETOK`∗-<`FORM`>, where `FORM` is replaced by the token string of the `FILE2` line), append placeholder characters (`?`) for the annotations expected from `FILE1`, append the corresponding `FILE2` lines.
- `-f` forced merge: mismatching `FILE2` tokens are merged with last `FILE1` token. This flag suppresses ∗`RETOK`∗ nodes, thus keeping the original token sequence intact. Annotations of merged lines are concatenated, using + as a separator.
- `-split` (boolean): `false` merges two CoNLL files and adopts the tokenization of the first. Tokenization mismatches from the second are represented by empty artificial tokens, i.e. words prefixed with ∗`RETOK`∗-... – `true` splits tokens from both files into maximal common substrings. From a split line, all annotations are copied to the lines of the subtokens. In order to mark the scope of a particular annotation, we use the I(O)BE(S) schema: Split annotations at the line of the first subtoken are prefixed by `B-` (begin), split annotations at the line of the last subtoken are prefixed by `E-` (end), and intermediate annotations are prefixed by `I-`. The flag `-split` is a shorthand for `-split=true`.

[4] `http://www.conll.org/`
[5] In the context of CoNLL shared tasks, one-word-per-line TSV formats have been applied to the following phenomena: shallow parsing (1999-2001), lexical semantics (2002-2003), dependency syntax (2006-2009, 2017-2018), semantic role labeling (2004-2005, 2008-2009), coreference (2011-2012), discourse parsing (2015-2016), inflectional morphology (2017-2018), applied NLP (2010, 2013-2014). Some recent shared tasks on highly abstract levels of linguistic analysis involved JSON formats along with the classical TSV format (discourse parsing, 2015-2016), or in their place (semantic parsing, 2019).
[6] In particular, on the number of columns, mismatches between parentheses, IOBES statements, cells without content and checks on comments.

- `-lev` perform Levenshtein alignment
- `-drop none` keep both versions of the merged column (by default, the `FILE2` column is dropped).

Additionally, we provide merge scripts for multiple (iterated) pair-wise alignments and final merging of multiple documents, as well as test cases for validity checks on the resulting CoNLL output.

4 Evaluation

We evaluate CoNLL Merge against two use cases: Alignment and annotation merging for multi-layer corpora, and alignment between deviating textual variants.

4.1 Same Text – Different Annotations

In order to evaluate the usefulness and practicability of *CoNLL Merge*, we describe a workflow of semantic annotation integration of a diverse collection of well-established, standard data sets which are all grounded on the *same* base texts taken from the Wall Street Journal (WSJ) data of the Penn Treebank [13, PTB]. We focus on the latest Penn Treebank version with syntactic phrase structure annotations [14, PTB3], PropBank [20, verbal predicate argument structure], NomBank [16, nominal semantic roles], OntoNotes [11, coreference], the Penn Discourse Treebank [21, PDTB2/shallow discourse structure], the RST Discourse Treebank [4, RSTDTB/hierarchical discourse structure], and the Discourse Graphbank [27, PDGB]. Crucially, not every resource provides annotations for every document in the PTB.

Figure 1 shows the (relative) number of corpora that a PTB file occurs in, averaged over WSJ sections. This information can be easily acquired by running *CoNLL Merge* on the distinct data sets as a preprocessing step. Ideally, a 100% bar in the chart would signify that each document of the respective section is annotated by all resources. Top sections range between 45–50% which indicates that an average document is found in half of the corpora.

It is important to note that all aforementioned resources come with partly conflicting, i.e. *varying* underlying tokenizations. The urgent need to make use of such multiple, distinct linguistic annotations in a *joint* learning framework (e.g., for discourse parsing based on syntactic dependencies, or semantic roles that are part of a coreference chain) has been the focus of a number of recent successful computational approaches [25, 26, 24, 23]. Figure 2 (left) illustrates our approach to harmonize concurrent tokenizations and to merge their annotation as part of a semantic annotation workflow that merges *all* levels of annotations provided for WSJ data.

In the first step, corpora with their original idiosyncratic tokenizations and linguistic annotations are converted to a CoNLL or TSV format. Some data sets provide CoNLL formats by default, for most others, converters are available. For the more exotic formats (PDGB, PDTB, RST-DTB), we provide CoNLL converters as part of the CoNLL Merge release. Sanity checks are performed by `CoNLLChecks`. Then, *pairwise* merges between two CoNLL files are produced (`CoNLLAlign`). Finally, *full* merges are generated resulting in the global data structure that shares the content of all base resources. In total, our method encounters 5,542 tokenization mitmatches out of which on average 98.7% are resolved and successfully merged with the different flag options.[7]

[7] Rare issues are encountered for cases in which subsequent tokenization conflicts appear immediately adjacent. Moreover, since merging can be easily parallelized, our routine runs reasonably fast ($<$ 3 mins

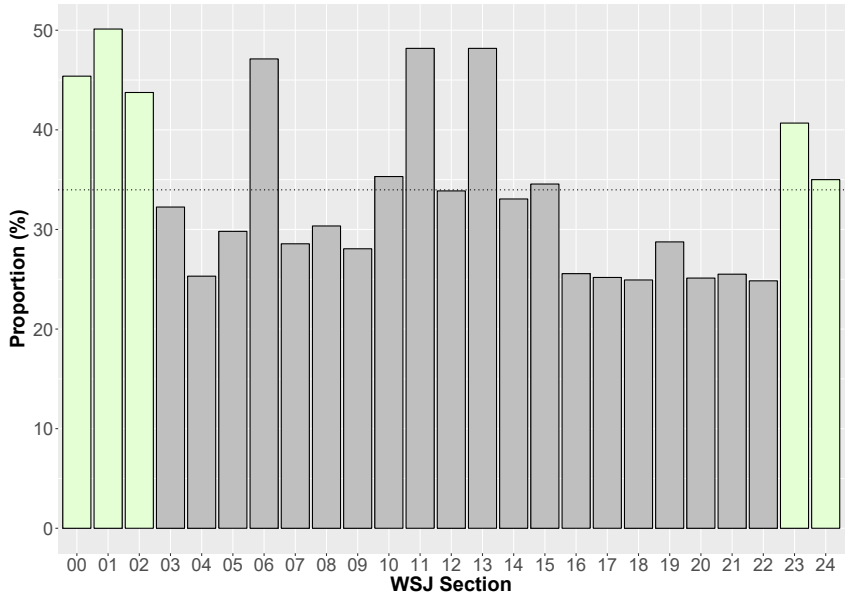


Figure 1 Relative number of corpora that contain a particular PTB document, averaged over WSJ sections.

Table 1 Alignment of OntoNotes (ON) parse files with other annotations, file wsj_0655, 992 (ON) tokens.

		PTB	RST	PDGB	PDTB*
default	1:1 ON alignment	959 (97%)	811 (82%)	834 (84%)	609 (61%)
alignment	1:0	33 (3%)	181 (18%)	158 (16%)	383 (39%)
	0:1	11 (1%)	113 (11%)	141 (14%)	51 (5%)
force	1:1 (no merge)	968 (98%)	839 (85%)	852 (86%)	643 (65%)
alignment	1:0	23 (2%)	114 (11%)	91 (9%)	340 (34%)
(-f)	0:1	0 (0%)	0 (0%)	0 (0%)	2 (0%)
	merged annotations	1 (0%)	39 (4%)	49 (5%)	9 (1%)
split	1:1 (no split)	959 (97%)	811 (82%)	834 (84%)	609 (61%)
alignment	1:0	0 (0%)	92 (9%)	91 (12%)	311 (31%)
(-split)	0:1	0 (0%)	32 (3%)	25 (9%)	12 (1%)
	split annotations	33 (3%)	50 (5%)	118 (12%)	98 (11%)
-split -f	1:1 (no split/merge)	959 (97%)	811 (82%)	834 (84%)	609 (61%)
	1:0	0 (0%)	72 (7%)	74 (7%)	270 (27%)
	0:1	0 (0%)	0 (0%)	0 (0%)	0 (0%)
	split/merged annotations	33 (3%)	62 (6%)	118 (12%)	98 (10%)

* contains text fragments only

Table 1 illustrates the extent of tokenization differences and the effect of the merging strategies for file `wsj_0655`, one of only seven files contained in the intersection of OntoNotes, Penn Treebank, RST Discourse Treebank, Penn Discourse Graphbank and Penn Discourse Treebank.[8] For illustration, we use the tokenization of OntoNotes parses as primary tokenization and match all other annotations against it. The PTB provides a slightly older version of the *same* annotations and tokenization, nevertheless deviating in 3% of the tokens. One source of deviation is in the treatment of hyphenized words and multi-word expressions. Note that these annotations do not just tokenize the original text. In addition to text tokens, empty tokens are inserted to represent syntactic movement operations. The RST edition provides untokenized text plus markup for paragraph boundaries. Similarly, the PDGB edition uses untokenized text. In the RST, PDGB and PDTB converters we provide, a TSV representation is created by treating every white-space separated string as a token. The PDTB edition is very different in that it is a standoff format which does not provide the full text, but only the text of the annotated spans, and their character offsets in the original text file. The standoff mechanism is defined with reference to a PTB version (similar but not identical to the OntoNotes version used here) or against the original plain text (which none of these corpora provide). Our converter does *not* attempt to resolve standoff references. Instead, we use the character offsets and the text provided in the span to *reconstruct* the plain text. As the spans contain only about 60% of the original text, this reconstruction is incomplete, its TSV representation can nevertheless be successfully aligned against OntoNotes (or any other full-text corpus).

As the table shows, the merging strategies have the following characteristics:

default is fully reversible, in that original token boundaries and the original annotations are preserved. The empty elements introduced for $0:m$ alignments, however, do interrupt the original sequence of tokens from `FILE1`.

-f enforces the tokenization of `FILE1` onto `FILE2`. `FILE2` annotations and tokenizations can be altered in an irreversible fashion, in that annotations are concatenated without the possibility to align them with their original token boundaries.

-split is fully reversible, in that the original token boundaries and the extent of the original can be recovered. Interruptions by $0:m$ alignments are minimized, but the original `FILE1` tokenization is altered. A tool expecting `FILE1` tokenization should not be applied to the output of this merging operation.

-split -f also enforces `FILE1` tokenization, but distributes `FILE2` annotations over multiple `FILE1` tokens (where applicable).

Three main objectives can be pursued: annotation reversibility (default), the establishment of a tokenization based on maximal shared substrings (-split, reduces both $n:0$ and $0:m$ alignments) or adoption of a privileged tokenization (-force, i.e., strictly enforce 1:1 or 1:0 alignments).

As an example, consider the following phrase from `wsj_0655`: *Lawmakers haven't publicly raised the possibility of renewing military aid[...]* Figure 2 (right) shows the result of three pairwise merges between the Penn Discourse Treebank as primary source and the annotations in PropBank. For illustration purposes, we first highlight the different tokenization outputs.

---

for the complete PTB with standard CPU).

[8] For demonstration purposes, the different versions of this file are included in the associated software distribution. However, for reasons of copyright, the archive is encrypted archive and the password must be requested from the first author. Alternatively, access to the different versions of the file can be requested from LDC, `https://catalog.ldc.upenn.edu`.

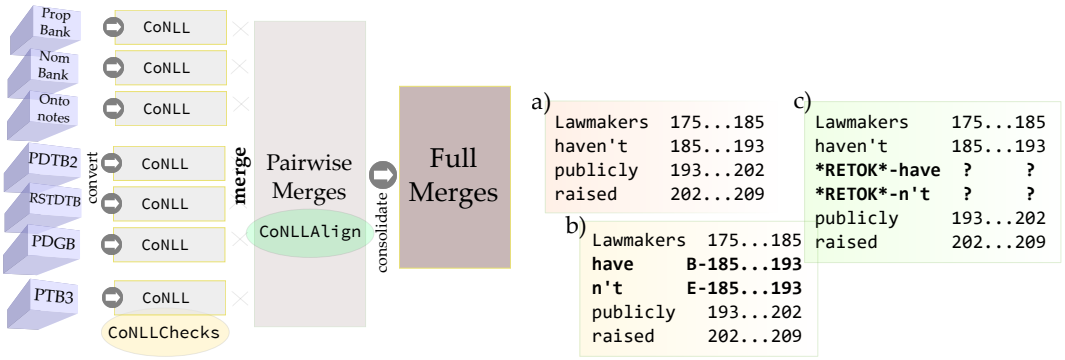


Figure 2

Left: Harmonizing PTB corpora (conversion & merging of ling. annotations) into one CoNLL output.

Right: Merged tokenizations between PDTB2 and PropBank: `-f` (a), `-split=true` (b), default (c).

Merging was performed with a combination of flag options (forced and default merging). The latter adopts the tokenization of the primary source, and inserts ∗RETOK∗ tokens, whenever alternatives are encountered within PropBank. With the `-split` option set to `true`, spans are equipped with underspecified beginning and end indices. The reason for having varying tokenizations across the two resources is due to the requirements of their idiosyncratic linguistic annotations. In PropBank, for instance, it is necessary to assign an individual modifier role to the negation (`ARGM-NEG`), therefore requiring a distinct token (*n't*) to be split from the orthographically combined auxiliary verb. In contrast, in the discourse setting of the PDTB2, only larger (shallow) spans are considered which dispense with the need for such a fine-grained segmentation. However, *CoNLL Merge* allows for a fruitful combination of both types of complementing linguistic annotations into one shared layer: Fig. 4 in Appendix A shows the combined information including discourse aspects as well as predicative argument structure (semantic role annotations) into one harmonized CoNLL token layer.

4.2 Same Source – Different Text

Beyond comparing and merging annotations of the same texts, CoNLL Merge can also be used as a collator for the alignment of different versions of the same text, and thus, for projecting annotations from one text to another. In philology, collation is the process of determining the differences between two or more variants of a text (e.g., different editions of a book, or different manuscripts of a particular text).

Designated tools for the purpose exist, e.g., CollateX [8], but they do currently not support the alignment of annotated text nor the projection of annotations from one textual variant to another. Instead, they focus on applications in stemmatology and the study of intertextual relations and provide graphical interfaces for the purpose. CollateX reads multiple plain text versions of a text, performs tokenization on each version, performs an alignment similar to that of CoNLL Merge and returns alignment results for further processing, for instance for producing a critical apparatus.

CoNLL Merge does provide a similar functionality, albeit with a focus on annotation rather than stemmatology and intertextuality:[9] We experimented with several merging routines on different Middle Low German editions of the *Interrogatio Sancti Anselmi de Passione Domini*. For reasons of copyright, we cannot ship the sample data, so we provide a script

[9] Scripts and data set available under `https://github.com/acoli-repo/conll/tree/master/data_phil`.

Table 2 Alignment of Anselmus ms. D with 6 other mss., 7263 tokens (D).

		D2	HA1521	Kh	O	SP	StA1495
default	1:1	6085 (84%)	4231 (58%)	4581 (63%)	4505 (62%)	4412 (61%)	3801 (52%)
alignment	1:0	993 (14%)	2856 (39%)	2506 (35%)	2575 (35%)	2675 (37%)	3286 (45%)
	0:1	1079 (15%)	2336 (36%)	2422 (35%)	2646 (37%)	2556 (37%)	2540 (40%)
Levenshtein	1:1	6817 (94%)	5453 (75%)	5831 (80%)	6070 (84%)	5900 (81%)	5186 (71%)
alignment	1:0	261 (4%)	1634 (22%)	1256 (17%)	1010 (14%)	1187 (16%)	1901 (26%)
(-lev)	0:1	347 (5%)	1114 (17%)	1172 (17%)	1081 (15%)	1068 (15%)	1155 (18%)

for downloading the original PDFs, for text extraction and for creating a CoNLL-compliant TSV file with three columns:

- We perform whitespace tokenization. Punctuation signs are not separated from grammatical words.
- The first columns contains the original string value, including punctuation signs.
- The second columns contains a normalized representation of the string value, with a number of orthographical conventions being harmonized (punctuation removed, lowercase, removal of diacritics). This normalization is not language-specific, but presupposes a Latin-based orthography.
- The third column contains a language-specific normalization of the normalized string. For instance, many medieval orthographies use *w*, *u* and *f* interchangeably with *v* (for different phonemes), so that *w,u,f* are all normalized to *v*.

For seven Middle Low German Anselmus manuscripts, we performed alignment (collation) over the third column. As gold data for the alignment of these texts is currently not available, we only report the number of successfully aligned tokens. Note that mis-alignment results in a low likelihood that subsequent tokens will be aligned, so that 1:1 alignments for more than 50% in default mode generally indicate alignment success. For random samples, this has been manually confirmed by the authors. However, the manuscripts differ considerable, both in their orthography and formulations, but also in additions and omissions. For instance, manuscripts D and D2 share a prolog of 235 tokens which is absent from the other manuscripts. Table 2 shows the alignment results for Anselmus ms. D. with six other manuscripts.

Figure 3 illustrates collation/alignment results for the second to fourth shared sentence of Anselmus mss. D, D2 and HA1521. This example illustrates typical alignment errors: Default alignment frequently fails to identify orthographic variants of the same word. These errors are inherited by force alignment, but not by Levenshtein alignment.

5 Summary

In this paper, we have described *CoNLL Merge*, a fully automated merging routine for harmonizing linguistic annotations in multi-layer corpora which are based on *concurrent tokenizations* of the same text.

To our best knowledge, CoNLL Mergeis the first system to perform this task in a generic fashion for one of the most popular corpus formalisms: one-word-per-line tab-separated values, as used in the CoNLL shared tasks, in popular corpus information systems [10] or for digital lexicography [12]. We are aware of existing implementations for handling conflicting tokenizations: Solutions based on hand-crafted rules [9] suffer from a lack of genericity. A number of libraries for merging CoNLL files do already exist, however, these are restricted to

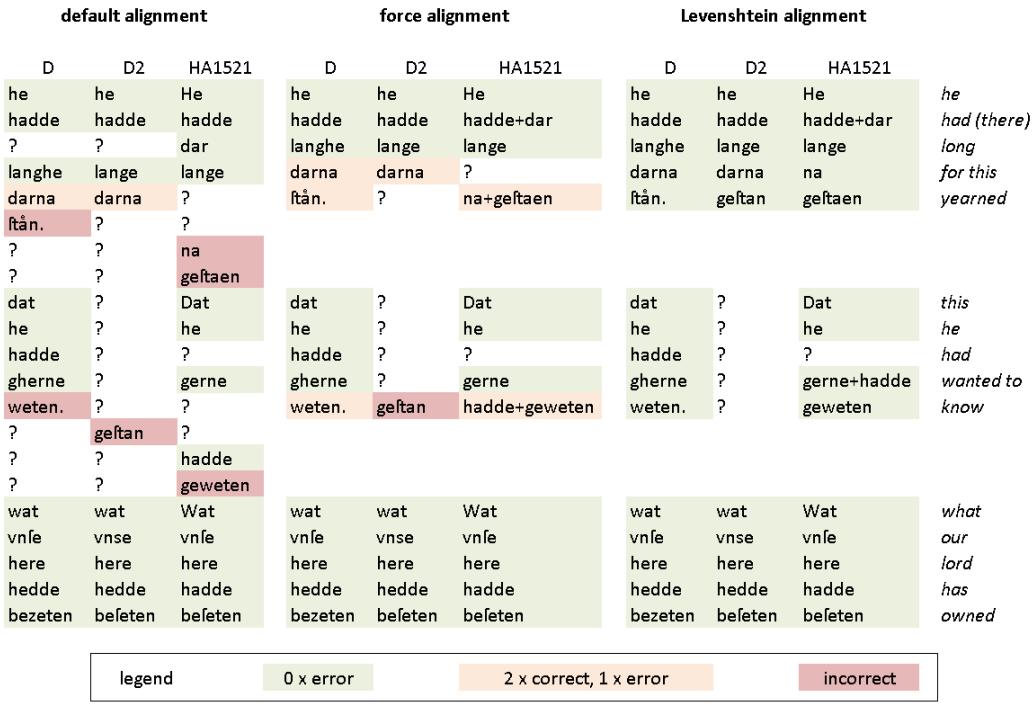


Figure 3 Collation results for Anselmus ms. D, D2 and HA1521, second to fourth shared sentence: *He yearned for this for a long time. He wanted to know: What did our lord own?* Question marks indicate the absence of an alignable token.

individual CoNLL dialects such as CoNLL-U[10] or CoNLL-X,[11] whereas our implementation is fully generic in that it allows the user to configure what column(s) to take as the basis for comparison. Our own earlier work on the automated resolution of tokenization mismatches [5] basically implemented the same functionality as CoNLL Merge, but this was closely tied to a highly complex standoff annotation formalism, not directly applicable to common corpus formats – and, effectively, forgotten. Finally, designated modules included in a number of NLP pipeline systems provide heuristic components for the resolution of tokenization mismatches, e.g., the Illinois NLP Curator [7] implements a maximum common substring strategy. These suffer from a similar limitation, i.e., these components are tightly integrated with a particular implementation, and not applicable to annotations in general. Moreover, we are not aware of any such module which allows the user to select among possible merging strategies.

Beyond this, we have shown that our implementation is applicable to texts with variation far beyond tokenization differences. In fact, CoNLL Merge can be used as a collator. Unlike other state-of-the-art collators, however, CoNLL Merge allows to perform collation with annotated texts and supports the projection of annotations from one text variant to another. For applications in NLP, this demonstrates that CoNLL Merge can also be applied for alignment of and annotation projection between paraphrases.

CoNLL Mergeis an efficient implementation of string-based comparisons, it works unsupervised, and does not require manual interference or any external resources. We demonstrated its applicability to successfully combine distinct linguistic annotations by connecting inform-

[10] `https://www.npmjs.com/package/conllu`
[11] `https://github.com/danieldk/conllx-utils`

ation from language resources which by default come with incompatible token layers. The augmented data obtained in this way enable improved insight into the interplay of annotations provided by distinct linguistic frameworks, allow for advanced NLP tool development, and due to its generic functionality could easily be extended to merging of morphologically more complex languages.

References

1 Thorsten Brants. TnT: A Statistical Part-of-speech Tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi:10.3115/974147.974178.

2 Lou Burnard. Reference guide for the British national corpus (XML Edition), 2007. URL: http://www.natcorp.ox.ac.uk/XMLedition/URGbnctags.html.

3 Jean Carletta, Stefan Evert, Ulrich Heid, Jonathan Kilgour, Judy Robertson, and Holger Voormann. The NITE XML Toolkit: Flexible annotation for multimodal language data. *Behavior Research Methods, Instruments, & Computers*, 35(3):353–363, 2003. doi:10.3758/BF03195511.

4 Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. RST Discourse Treebank, 2002. LDC Catalog No.: LDC2002T07, ISBN, 1-58563-223-6.

5 Christian Chiarcos, Julia Ritz, and Manfred Stede. By all these lovely tokens... Merging Conflicting Tokenizations. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 35–43, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/W/W09/W09-3005.

6 Christian Chiarcos, Julia Ritz, and Manfred Stede. By all these lovely tokens... Merging conflicting tokenizations. *Language resources and evaluation*, 46(1):53–74, 2012.

7 James Clarke, Vivek Srikumar, Mark Sammons, and Dan Roth. An NLP Curator (or: How I Learned to Stop Worrying and Love NLP Pipelines). In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC-2012)*, Istanbul, Turkey, 2012. ELRA.

8 R. H. Dekker and G. Middell. Computer-Supported Collation with CollateX: Managing Textual Variance in an Environment with Varying Requirements. In *2nd Conference on Supporting Digital Humanities 2011 (SDH-2011)*, University of Copenhagen, Denmark, 2011.

9 Rebecca Dridan and Stephan Oepen. Tokenization: Returning to a Long Solved Problem. A Survey, Contrastive Experiment, Recommendations, and Toolkit. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 378–382, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P12-2074.

10 Stefan Evert and Andrew Hardie. Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium. In *Proceedings of Corpus Linguistics 2011 (CL2011)*, University of Birmingham, 2011.

11 Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. OntoNotes: The 90% Solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57–60, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=1614049.1614064.

12 Adam Kilgarriff, Vít Baisa, Jan Bušta, Miloš Jakubíček, Vojtěch Kovář, Jan Michelfeit, Pavel Rychlỳ, and Vít Suchomel. The Sketch Engine: Ten years on. *Lexicography*, 1(1):7–36, 2014.

13 Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi:10.3115/1075812.1075835.

14 Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. Treebank-3, 1999. LDC Catalog No.: LDC99T42, ISBN, 1-58563-163-9.

15 Edward M. McCreight. A Space-Economical Suffix Tree Construction Algorithm. *J. ACM*, 23(2):262–272, 1976. doi:10.1145/321941.321946.

16 Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. The NomBank Project: An Interim Report. In *In Proceedings of the NAACL/HLT Workshop on Frontiers in Corpus Annotation*, 2004.

17 T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at International Conference on Learning Representations*, 2013.

18 Christoph Müller and Michael Strube. Multi-level annotation of linguistic data with MMAX2. In Sabine Braun, Kurt Kohn, and Joybrato Mukherjee, editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang, Frankfurt a.M., Germany, 2006.

19 Eugene W. Myers. AnO(ND) difference algorithm and its variations. *Algorithmica*, 1(1):251–266, 1986. doi:10.1007/BF01840446.

20 Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Comput. Linguist.*, 31(1):71–106, March 2005. doi:10.1162/0891201053630264.

21 Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank 2.0. In *Proceedings, 6th International Conference on Language Resources and Evaluation*, pages 2961–2968, Marrakech, Morocco, 2008.

22 James Pustejovsky, Adam Meyers, Martha Palmer, and Massimo Poesio. *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, chapter Merging PropBank, NomBank, TimeBank, Penn Discourse Treebank and Coreference, pages 5–12. Association for Computational Linguistics, 2005. URL: http://aclweb.org/anthology/W05-0302.

23 Michael Roth. Role Semantics for Better Models of Implicit Discourse Relations. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*, 2017. URL: http://aclweb.org/anthology/W17-6934.

24 Niko Schenk, Christian Chiarcos, Kathrin Donandt, Samuel Rönnqvist, Evgeny Stepanov, and Giuseppe Riccardi. Do We Really Need All Those Rich Linguistic Features? A Neural Network-Based Approach to Implicit Sense Labeling. In *Proceedings of the CoNLL-16 shared task*, pages 41–49. Association for Computational Linguistics, 2016. doi:10.18653/v1/K16-2005.

25 Carina Silberer and Anette Frank. Casting Implicit Role Linking as an Anaphora Resolution Task. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 1–10, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=2387636.2387638.

26 Mihai Surdeanu, Tom Hicks, and Marco Antonio Valenzuela-Escárcega. Two Practical Rhetorical Structure Theory Parsers. In *HLT-NAACL*, pages 1–5. The Association for Computational Linguistics, 2015.

27 Florian Wolf, Edward Gibson, Amy Fisher, and Meredith Knight. Discourse Graphbank, 2005. LDC Catalog No.: LDC2005T08, ISBN, 1-58563-320-8.

28 Kaoru Yamamoto, Taku Kudo, Akihiko Konagaya, and Yuji Matsumoto. Protein Name Tagging for Biomedical Annotation in Text. In Sophia Ananiadou and Jun'ichi Tsujii, editors, *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 65–72, 2003.

29 Li Yujian and Liu Bo. A Normalized Levenshtein Distance Metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1091–1095, June 2007. doi:10.1109/TPAMI.2007.1078.

A Merging Discourse and SRL Structure – Sample CoNLL Merge

The following figure illustrates the merging of discourse and verbal predicate argument structure annotations of the same texts from two distinct resources. The resulting CoNLL format contains columns for tokens, character begin and end offsets, discourse meta data (blue), phrase structure (green), semantic roles (purple). Note, that *haven't* is treated as a single token in the PDTB2. The resulting output below contains *two* tokens with partial production rules (green) assigned to them.

```
Lawmakers  175...185  1:Arg1  (Explicit and,  Expansion.Conjunction);
                      3:Arg1  (Explicit but,  Comparison.Contrast.Juxtaposition)
                      NNS ( (S (S (NP-SBJ *)  _ ARG0 _ ARG0

have      B-185...193 B-1:Arg1 (Explicit and, Expansion.Conjunction);
                      3:Arg1   (Explicit but, Comparison.Contrast.Juxtaposition)
                      S-VBP (VP *             _ _ _ _

n't       E-185...193 E-1:Arg1 (Explicit and, Expansion.Conjunction);
                      3:Arg1   (Explicit but, Comparison.Contrast.Juxtaposition)
                      S-RB *                  _ S-ARGM-NEG _ _

publicly   193...202  1:Arg1   (Explicit and, Expansion.Conjunction);
                      3:Arg1   (Explicit but, Comparison.Contrast.Juxtaposition)
                      RB (VP ( ADV-MNR *)     _ ARGM-MNR _ _

raised     202...209  1:Arg2   (Explicit and, Expansion.Conjunction);
                      3:Arg1   (Explicit but, Comparison.Contrast.Juxtaposition)
                      VBN *                   raise.v.01 rel _ _
...
                                                          merged CoNLL
```

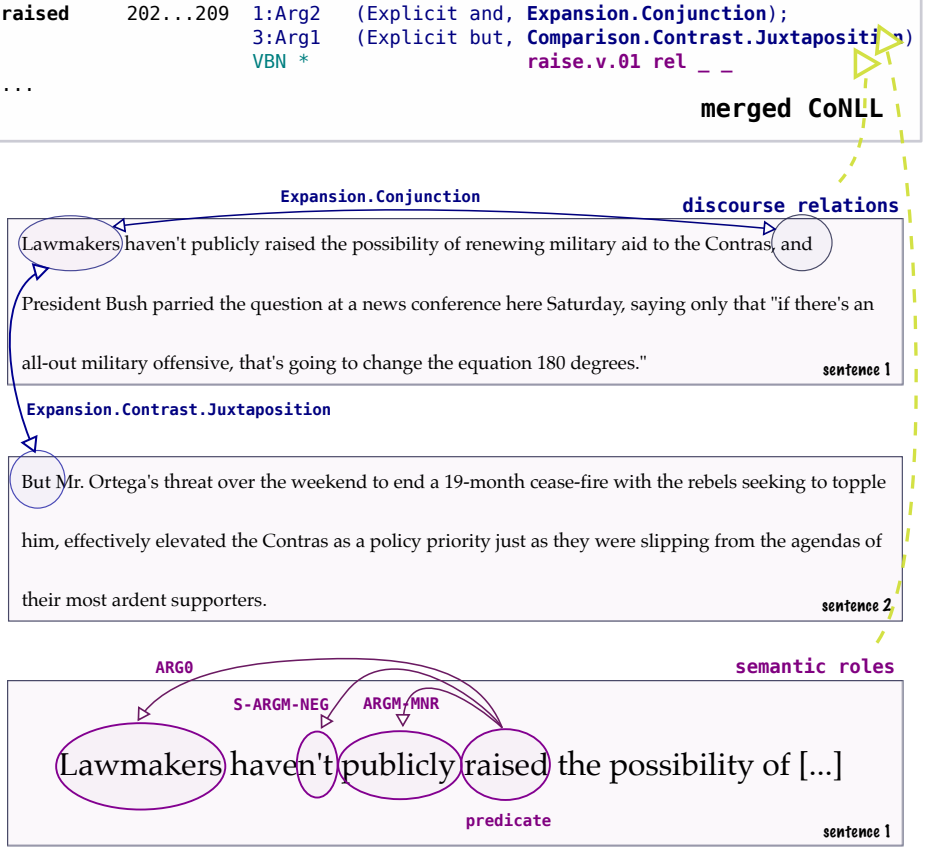


Figure 4 Merged CoNLL annotations (top) between PTB base source, PDTB2, and PropBank. The resulting output combines both phrase structure, **discourse structure** (blue) and **semantic roles** (purple). One word per line format expanded for better legibility.

Exploiting Background Knowledge for Argumentative Relation Classification

Jonathan Kobbe
University of Mannheim, Germany
jonathan@informatik.uni-mannheim.de

Juri Opitz
Heidelberg University, Germany
opitz@cl.uni-heidelberg.de

Maria Becker
Heidelberg University, Germany
mbecker@cl.uni-heidelberg.de

Ioana Hulpuş
University of Mannheim, Germany
ioana@informatik.uni-mannheim.de

Heiner Stuckenschmidt
University of Mannheim, Germany
heiner@informatik.uni-mannheim.de

Anette Frank
Heidelberg University, Germany
frank@cl.uni-heidelberg.de

Abstract

Argumentative relation classification is the task of determining the type of relation (e.g., SUPPORT or ATTACK) that holds between two argument units. Current state-of-the-art models primarily exploit surface-linguistic features including discourse markers, modals or adverbials to classify argumentative relations. However, a system that performs argument analysis using mainly rhetorical features can be easily fooled by the stylistic presentation of the argument as opposed to its content, in cases where a weak argument is concealed by strong rhetorical means. This paper explores the difficulties and the potential effectiveness of knowledge-enhanced argument analysis, with the aim of advancing the state-of-the-art in argument analysis towards a *deeper, knowledge-based understanding and representation of arguments*.

We propose an argumentative relation classification system that employs linguistic as well as knowledge-based features, and investigate the effects of injecting background knowledge into a neural baseline model for argumentative relation classification. Starting from a Siamese neural network that classifies pairs of argument units into SUPPORT vs. ATTACK relations, we extend this system with a set of features that encode a variety of features extracted from two complementary background knowledge resources: ConceptNet and DBpedia. We evaluate our systems on three different datasets and show that the inclusion of background knowledge can improve the classification performance by considerable margins. Thus, our work offers a first step towards effective, knowledge-rich argument analysis.

2012 ACM Subject Classification Computing methodologies → Neural networks; Information systems → Graph-based database models; Information systems → Clustering and classification

Keywords and phrases argument structure analysis, background knowledge, argumentative functions, argument classification, commonsense knowledge relations

Digital Object Identifier 10.4230/OASIcs.LDK.2019.8

Funding This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the project ExpLAIN, Grant Number STU 266/14-1 and FR 1707/-4-1, as part of the Priority Program

© Jonathan Kobbe, Juri Opitz, Maria Becker, Ioana Hulpuş, Heiner Stuckenschmidt, and Anette Frank;
licensed under Creative Commons License CC-BY

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 8; pp. 8:1–8:14

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

"Robust Argumentation Machines (RATIO)" (SPP-1999) as well as by the Leibniz Science Campus Empirical Linguistics & Computational Language Modeling, supported by Leibniz Association grant no. SAS2015-IDS-LWC and by the Ministry of Science, Research, and Art of Baden-Württemberg.

Acknowledgements We gratefully acknowledge the support of NVIDIA Corporation through a donation of GPUs that were used for this research.

1 Introduction

Attack and *support* are two important relations that can hold between argumentative units. Consider the following two argumentative units **(1)** and **(2)** that are given in response to the topic **(0)** *Smoking should be allowed in every restaurant*:

(1) *Smoking is a significant health hazard.*
(2) *Combustion processes always produce toxins.*

Both **(1)** and **(2)** have a negative *stance* towards the topic **(0)**, and at the same time they stand in a *support* relation themselves: (**2**) *supports* (**1**). In textual discourse, this relationship is often indicated with discourse markers, e.g., *because* (i.e., **(1)** *because* **(2)**), or *therefore* (i.e., **(2)**, *therefore* **(1)**). Similarly, *attack* relations are frequently marked with discourse markers, e.g., *A, however, B*, etc. Although in the given example, the argumentative units **(1)** and **(2)** have no words in common and do not include discourse markers, a human can easily determine the *support* relation between them. This can be done for instance by recognizing relations that connect the two units like the fact that *smoking generally involves a combustion process* and that *toxins are detrimental to health.*

While accessing such knowledge is seamless for humans, it is much more challenging for machines. State-of-the-art machine learning systems for argument analysis (for instance [27] or [1]) mainly rely on the exploitation of shallow linguistic markers (such as adverbials, discourse connectors or punctuation) and largely ignore background knowledge and common sense reasoning as evidences for classifying argumentative relations. We argue that for building reliable systems, world knowledge and common sense reasoning should be core criteria and evidences for determining whether an argumentative unit A attacks or supports B. Rather than solving the argumentative relation classification or argumentation structure reconstruction task by using only linguistic indicators that characterize the *rhetorics* of the argument, we emphasize the need of systems that are able to capture the underlying logics of an argument by analyzing its content.

Clearly, this is a challenging task, as it requires appropriate knowledge sources and reasoning capacities. However, exploiting the knowledge relations that hold between argument units carries an immense potential of *explaining, in interpretable ways,* why an argument holds (or does not hold), when presenting supporting or attacking evidence. We therefore use the opportunity brought by the current advances in the Linked Open Data movement, and investigate the potential of external, structured knowledge bases such as ConceptNet and DBpedia, for providing the required background knowledge. Specifically, we propose a series of knowledge-based features for argumentative relation classification and analyze their impact as compared to surface-linguistic features as used in current state-of-the-art models. Starting with a linear regression classifier, we proceed to a stronger Siamese neural network system that encodes pairs of argumentative units to classify their relation. This system, when enriched with knowledge-based features, yields considerable performance improvements over the non-enriched version, and thus offers clear indications for the prospects of knowledge-enhanced argument structure analysis.

Our contributions are as follows: (i) we propose features that extract background knowledge from two complementary knowledge resources: ConceptNet and DBpedia and analyze their respective impact on the task; (ii) we show that a neural system enriched with background knowledge obtains considerable performance gains over the non-enriched baseline. In sum, our work is one of the first to shows positive impact of background knowledge on argument classification.

2 Related Work

2.1 Argument Structure Analysis

Stab and Gurevych, (2014) [26] propose an approach for (1) identifying argument components and (2) classifying the relation between pairs of argument components as either supportive or non-supportive. They propose several features, including structural features (e.g. number of tokens of the argument component, token ratio between covering sentence and argument component), lexical features (n-grams, verbs, adverbs, modals), syntactic features (e.g. production rules as proposed by [13]), contextual features (e.g. number of punctuations and number of tokens of the covering sentence), and further indicators such as discourse markers and pronouns, which are fed into a SVM classifier. When trained on the corpus of student essays that we also use in this work [25], the system obtains F1-scores of up to 0.726 for identifying argument components and 0.722 for distinguishing support from non-support relations. Following up the task of argument structure analysis, Stab and Gurevych, (2017)[27] propose an end-to-end approach where they first identify argument components using sequence labeling at the token level. For detecting argumentation structures, they then apply a model which jointly distinguishes argument component types (major claim, claim, premise) and argumentative relations (linked vs. not linked) using Integer Linear Programming. Finally, the stance recognition model differentiates between support and attack relations using a SVM classifier with lexical, sentiment, syntactic and structural features (similar to the features used in their previous work [26]) as well as PDTB discourse relations and combined word embeddings. They evaluate their model on the student essay corpus and the Microtext corpus [19], achieving F1 scores of 0.68 and 0.75 respectively on the task of stance classification (support vs. attack). Similar to Stab and Gurevych [26, 27], Persing and Ng (2016) [21] propose an End-to-End system for identifying argument components and the relations that occur between them in the student essay corpus. Their baseline system is a pipeline which first extracts argument components heuristically and then distinguishes firstly between argumentative and non-argumentative spans and subsequently between attack vs. support vs. not related relations. For both classifiers they apply maximum entropy classification, using the same features as Stab and Gurevych [26, 27]. This baseline system is outperformed by a joint model which uses global consistency constraints to perform joint inference over the outputs of the single pipeline tasks in an ILP framework, achieving F1 scores of up to 38.8% for the relation identification task.

The features used in these approaches are partly also used in our Baseline system (e.g. sentiment, token and punctuation statistics, modal verbs). Nonetheless, in this work we take a step further, by leveraging external knowledge bases such as DBpedia and ConceptNet in addition to our linguistic feature set.

Nguyen and Litman (2016)[16] also address the task of argumentative relation classification based on the student essay corpus. They adapt Stab and Gurevych's (2014) [26] system by adding contextual features extracted from surrounding sentences of source and target

components as well as from topic information of the writings. For identifying attack relations, they achieve up to 0.33 F1 scores, and for support relations 0.94 F1 scores, which shows that contextual features are helpful for the task of relation classification. In contrast, we aim for an approach that is agnostic of the context in which the argument units originally occur.

Most existing work on argument analysis focuses on classifying relations between argument units in monologic argumentation, partly due to the used /available datasets. Since our aim is to assess pairs of argument units regardless of whether they belong to the same monologue, we create a new dataset, sourcing pairs of argumentative units from Debatepedia[1]. In this regard, our work is comparable to Hou and Jochim's (2017) [9], who learn to predict for pairs of argument units stemming from different texts in Debatepedia whether they are in agreement or disagreement with each other. They apply various models including an attention-based LSTM, a textual entailment system, and classification models trained by logistic regression. Their best performing system utilizes the mutual support relations between argumentative relation classification and stance classification jointly and achieves an accuracy of 65.5%, which confirms that there is a close relationship between argumentative relation classification and stance classification.

The relation between our task of argumentative relation classification and the task of stance classification has also been discussed by Peldszus and Stede (2015) [18] and by Afantenos et. al (2018) [1]. Compared to the binary distinction (support vs. attack) in our work and in Hou and Jochim (2017) [9] (agree vs. disagree), the annotation of their argumentation structure is more fine-grained and contains several aspects. The structure follows the scheme outlined by Peldszus and Stede (2013) [17], where the different aspects are (1) finding the central claim of the text, (2) predicting the relation between that claim and the other segments, (3) predicting the relation between the other segments, (4) identifying the argumentative role of each segment, and (5) predicting the argumentative function of each relation. Similar to Hou and Jochim (2017) [9], they show that joint predictions - in this case the prediction of all these levels in the evidence graph - help to improve the classification on single levels.

Menini and Tonelli (2016) [15] also address the task of distinguishing agreement vs. disagreement relations of argument components in a dialogic setting, investigating documents from political campaigns and Debatepedia. They introduce three main categories of features: sentiment-based features (e.g. the sentiment of the statements and sentiment of the topic), semantic features (e.g word embeddings, cosine similarity and entailment), and surface features (e.g. the lexical overlap and the use of negations). Using all features jointly as input to an SVM classifier, they achieve up to 83 % accuracy on the political campaign dataset and 74 % accuracy on Debatepedia.

2.2 Background Knowledge for Argument Analysis

External knowledge resources have been leveraged as supporting information for various tasks in NLP, including Argument Analysis. Potash et al. (2017) [22] assess the feasibility of integrating Wikipedia articles when predicting convincingness of arguments and find that they can provide meaningful external knowledge. Habernal et al. (2018) [7] claim that comprehending arguments requires significant language understanding and complex reasoning over world knowledge, especially commonsense knowledge. Incorporating external knowledge is therefore viewed as essential for solving the SemEval Argument Reasoning Comprehension

[1] `http://www.debatepedia.org`

Task (2018 Task 12, [7]) [2]. This can be confirmed by the results of the participating systems: The best performing system, proposed by Choi and Lee [6], is a network transferring inference knowledge to the argument reasoning comprehension task. It makes use of the SNLI dataset [4] and benefits from similar information in both datasets. This system outperforms all other systems by more than 10%. Besides pretrained word embeddings (e.g. contexualized embeddings, [11]) and a sentiment polarity dictionary [5], none of the other published systems takes into account external knowledge resources for solving the task.

Following up on the observation about the usefulness of external knowledge for argumentative reasoning, the approach of Botschen et al. (2018) [3] leverages event knowledge from FrameNet and fact knowledge from Wikidata to solve the Argument Reasoning Comprehension task. They extend the baseline model of Habernal et al. (2018) [7], an intra-warrant attention model that only uses conventional pretrained word embeddings as input, with embeddings for frames and entities derived from FrameNet and Wikipedia, respectively. They conclude that external world knowledge might not be enough to improve argumentative reasoning. However, motivated by the promising results of Becker et al. 2017 [2] who have shown that commonsense knowledge that is useful for understanding Microtext arguments can be mapped to relation types covered by ConceptNet, we analyze additional knowledge bases, specifically ConceptNet for commonsense knowledge and DBpedia for world knowledge.

3 Knowledge Graph Features

For exploiting background knowledge, we designed features based on two knowledge graphs: ConceptNet [3] and DBpedia [4]. We expect ConceptNet to contain valuable information about common sense knowledge while DBpedia captures encyclopedic knowledge. The core idea is to connect pairs of argumentative units via relations in the knowledge graphs and to use the relation types and the extracted paths as features. The intuition is that certain types of paths or relations, like e.g. the *Antonym* relation in ConceptNet, occur more often in disagreeing and therefore attacking pairs of statements than in supporting ones and vice versa.

Given two argumentative units, we first proceed to link them to the external knowledge bases. Section 5.2 provides the entity linking details. Once the two argumentative units are linked, we represent them as sets A and B of their linked entities. We then pair all the elements in A to those in B. For each such pair $(x, y) \in A \times B, x \neq y$, we extract all the paths from x to y up to length three within the knowledge base. Figure 1 shows a graph consisting of such paths extracted from ConceptNet. As one can see in the graph, each path consists of nodes connected by directed edges labeled with *relation types*. As mentioned above, we assume that those relation types contain valuable information. For that reason, we design two kinds of features that rely on them: First, we check how often a certain relation type occurs along all paths between all pairs $(x, y) \in A \times B, x \neq y$ and divide that number by the total count of edges. This way, each relation type is a numerical feature on its own and all those features together sum up to 1. Second, we specifically exploit the paths. Since there are too many paths to create one feature per path, we group them via patterns. Each pattern is a multiset of relation types. For example, given the pattern [*Synomym,RelatedTo,RelatedTo*], the graph in Figure 1 contains two paths between *math*

[2] Given an argument consisting of a claim and a reason, the task is to select one out of two potential inferential licenses, called warrants, that explains the reasoning underlying the argument.
[3] `http://conceptnet.io`
[4] `http://dbpedia.org`

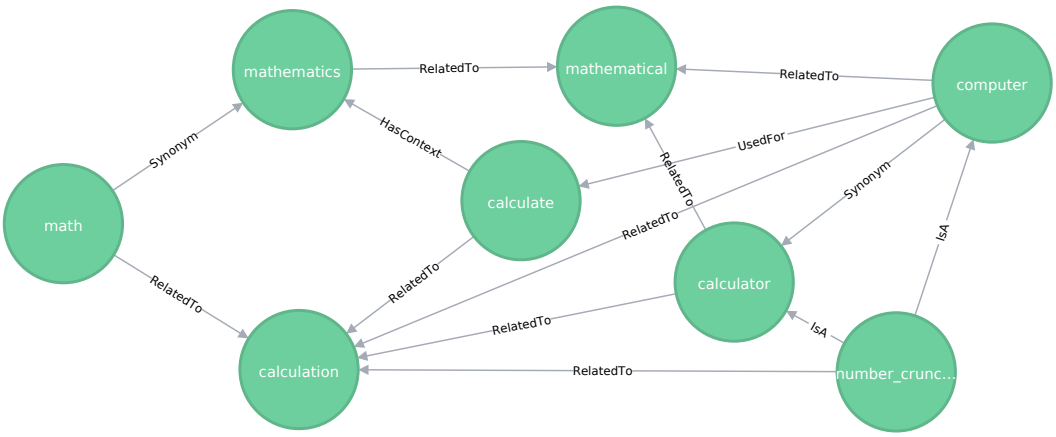


Figure 1 Connection between *math* and *computer* in ConceptNet, generated using Neo4j[5].

and *computer* that instantiate this pattern:

$$math \xrightarrow{Synonym} mathematics \xrightarrow{RelatedTo} mathematical \xleftarrow{RelatedTo} computer$$

$$math \xrightarrow{RelatedTo} calculation \xleftarrow{RelatedTo} calculator \xleftarrow{Synonym} computer$$

Each such path pattern corresponds to a numerical feature whose value is the number of its instantiations divided by the total number of paths. As some of the relation type-based and path-based features described above occur only rarely, we only use those features that occur in at least one percent of all the instances in the training data.

Besides exploiting the relation types and paths, we also hypothesize that the length and number of paths are useful for classification, as they provide an indication to the relatedness of A and B [10]. To account for this, we additionally compute (i) one feature representing the total number of paths divided by $|A| \cdot |B|$, (ii) three features representing the number of paths of a certain length i ($i \in \{1, 2, 3\}$) divided by the total number of paths, (iii) one feature representing the total number of identical entities in A and B divided by $|A| \cdot |B|$ and (iv) one feature with the count of all the different nodes along all paths divided by $|A| \cdot |B|$ again.

4 Neural Network Model

We design a Siamese neural network model for argumentative relation classification (**NN**). The architecture of the model is displayed in Figure 2. It consists of one Bi-LSTM [8], which is used to embed two argumentative units A and B into a common vector space. More precisely, sequences of word embeddings[6], $(e(w_1^A), ..., e(w_n^A))$ and $(e(w_1^B), ..., e(w_m^B))$ are fed through the Bi-LSTM to induce representations $emb(A), emb(B) \in \mathbb{R}^{2h}$, where h is the number of the two LSTM's hidden units (we concatenate the last states of the forward and backward pass of each LSTM). Based on the argument representations $emb(A)$ and $emb(B)$ we then compute a representation for the relation holding between these units by computing the difference vector between their representations $emb(A)$ and $emb(B)$: $r(A, B) =$

[5] `https://neo4j.com/`
[6] we use pre-trained 300d Glove vectors [20].

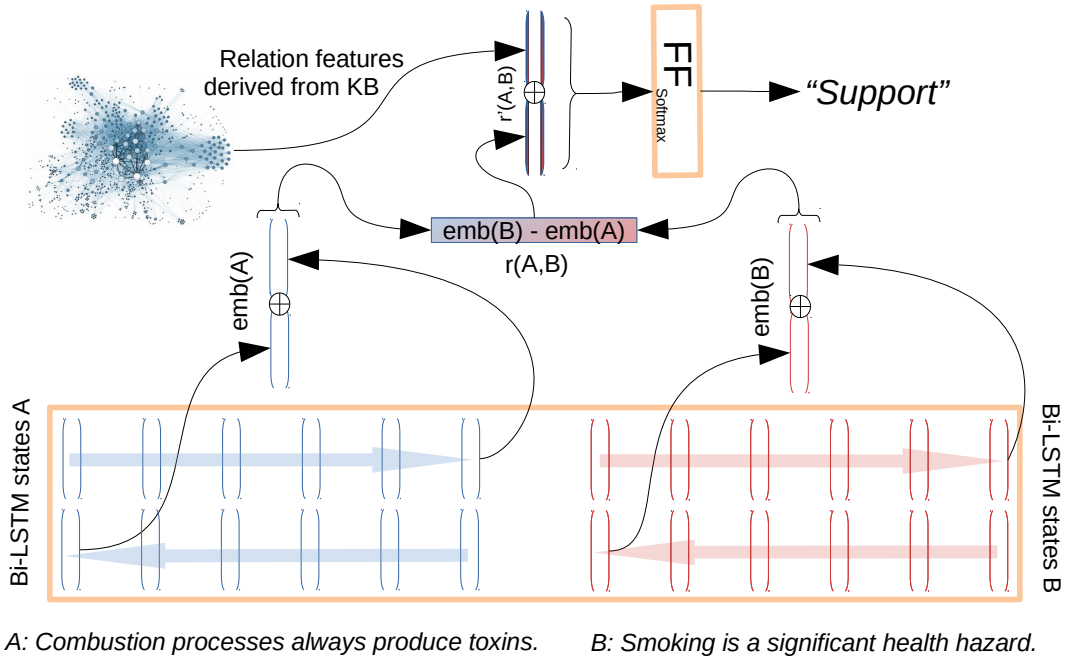


Figure 2 Architecture of the Siamese neural argumentative relation classifier. After embedding the argumentative units, their relation is defined as the vector offset between the unit representations in argument space. This representation can be enriched with a feature vector derived from background knowledge sources (e.g., ConceptNet).

$emb(B) - emb(A)$. The obtained representation for the relation can be further enriched by adding, e.g., features extracted from an external knowledge base that represent relevant information about knowledge relation paths connecting concepts and entities mentioned in the two argumentative units (cf. Section 3 and *relation features derived from KB*, Figure 2). The vector $v_K(A, B)$ that encodes such knowledge features is concatenated to the argument relation vector $r(A, B)$ to yield the extended vector representation $r'(A, B)$ of the argumentative relation: $r'(A, B) = r(A, B) \oplus v_K(A, B)$, where $x \oplus y$ denotes concatenation of vectors x, y. This final relation representation is further processed by a fully connected feed-forward layer (*FF*, Figure 2) with two output units and softmax-activations for providing the *support* and *attack* probabilities.

5 Experiments

We conduct experiments on three argumentative data sets from different domains, which will be described in the following section. Because we want the models to focus on the background knowledge involved in the argumentation, we consider only the argumentative units without their context and position. This increases the difficulty of the task as models are prevented from exploiting contextual and positional features.

Table 1 Data statistics for the different experimental datasets.

	Debatepedia	Microtexts	Student essays (Essays)
Total number of relations	14,441	308	1,473
Number of attack relations	7,184	84	161
Number of support relations	7,257	224	1,312

5.1 Data

Student Essays (Essays). The student essays consist of 90 persuasive essays in the English language. The essays were selected from *essayforum*[7] and annotated by [25]. The corpus contains 1473 annotated argumentative relations: 1312 were labeled as *support* and the remaining 161 were labeled as *attack* relations. We apply the same split between training and test data as [26] and [16]. For our purpose, we make use of pairs of attacking and supporting argumentative units and dismiss all other information about the position and context and the annotated argumentative components and stances.

Microtexts. This corpus consists of 112 short argumentative texts [19]. The corpus was created in German and has been translated to English. We use only the English version. The corpus is annotated with argumentation graphs where the nodes are argumentative units and the edges are argumentative functions. We again collect pairs of attacking and supporting argumentative units. Therefore, we consider only direct connections between two argumentative units that are labeled as *support* or *rebut*. We deliberately ignore the *undercut* function as an undercut is an attack on the argumentative relation between two argumentative units. This way, we extract 308 argumentative relations whereof 224 are support and 84 are attack relations. To achieve a proper split between training and testing data, we use all the Microtexts about *public broadcasting fees on demand, school uniforms, increase weight of BA thesis in final grade* and *charge tuition fees* for testing and all the others for training.

Debatepedia. This is a website where users can contribute to debates on some specific topic [8]. Most debates consist of a title, a topic that is formulated as a polar question (e.g. *Should the legal age for drinking alcohol be lowered?*), subtopics and arguments that are either in favor or against the topic. We crawled the Debatepedia website and extracted all arguments with a valid URL. In many arguments, the argument's claim is highlighted, so we used this feature to identify the claims, and removed the arguments that did not have any highlighted text. This resulted in 573 debates. We generate the pairs of argument units by pairing the topic of the debate to the claim. If the argument is in favor of the topic, then its claim *supports* the topic, else it *attacks* the topic. This way, we generate a large corpus containing 14441 pairs of argument units whereof 7257 are in support and 7184 are in attack relations. We arbitrarily chose 114 (20%) out of the 573 debates for testing and use the rest for training [9].

[7] `https://essayforum.com/`
[8] `http://www.debatepedia.org/`
[9] For information about accessing the data, see `http://explain.cl.uni-heidelberg.de/`.

5.2 Knowledge Graphs

DBpedia.[10] This knowledge graph contains information from Wikipedia[11] in a structured way. The English version contains more than 4 million entities classified in an ontology. For our work with DBpedia, we included the following datasets in English version in addition to the DBpedia Ontology (Version 2016-10): *article categories, category labels, instance types, labels, mapping-based objects* and *SKOS categories*. To achieve less meaningless paths, we excluded all the resources whose URI starts with *Category:Lists_of, List_of, Glossary_of, Category:Glossaries_of, Images_of, Category:Indexes_of, Category:Outlines_of, Category:Draft-Class, Category:Wikipedia* as well as the resource *owl:Thing*. For linking tokens in the argumentative units to entities in DBpedia, we use DBpedia Spotlight[12] with a minimum confidence of 0.3 and support of 1.

ConceptNet.[13] ConceptNet is a crowd-sourced resource of commonsense knowledge created by the Open Mind Common Sense (OMCS) project [23], to which were later added expert-created resources [24]. It has been built in response to the difficulties of automatic acquisition of commonsense knowledge. The current version, ConceptNet 5.6, comprises 37 relations, some of which are commonly used in other resources like WordNet (e.g. *IsA, PartOf*) while most others are more specific to capturing commonsense information and as such are particular to ConceptNet (e.g. *HasPrerequisite* or *MotivatedByGoal*). We use the English version of ConceptNet 5.6 which consists of 1.9 million concepts and 1.1 million links to other databases like DBpedia for instance. We deleted all self-loops as they don't contain any valuable information. Linking of tokens to ConceptNet is done in a straightforward way: We split the argumentative unit into maximum-length sequences of words that can be mapped to concepts. If a concept consists only of stop words or has a degree of less then three, it is dismissed [14]. This way, unconnected and only weakly connected concepts are avoided. If a concept consists of a single word, we use Stanford CoreNLP ([14]) to find out whether this is an adjective, noun or verb, in order to link it to the appropriate concept in ConceptNet, if possible.

5.3 Baselines

In this paper, we focus on *local* argumentative relation classification, thus our work is not directly comparable to prior work which proposes *global*, i.e., contextually aware classifiers for this task [26, 16, 18]. More specifically, we are interested in a classification setup that is agnostic of the contextual surface features such as discourse markers and position in discourse, and that restricts classification to the analysis of two argumentative units combined with the background knowledge that connects them.

Nevertheless, in order to compare to knowledge-lean paradigms of related work, we replicate features used in the most related previous work [26, 15]. To this end, we train a linear classifier with the replicated (linguistic) features, which we denote as **Ling**. As **Ling** features we use the sentiment of both argumentative units as features, as described in [15]. We simplified the negation features of [15] and use Stanford CoreNLP ([14]) to only

[10] `https://wiki.dbpedia.org/`
[11] `https://www.wikipedia.org/`
[12] `https://www.dbpedia-spotlight.org/`
[13] `http://conceptnet.io/`
[14] We use the default stopwordlist from `https://www.ranks.nl/stopwords` including *can*.

recognize whether there is some negation in an argumentative unit. From [26] we adopted the structural features which contain token and punctuation statistics and two features indicating whether a modal verb occurs. Additionally, we use each pair of words, one from each argumentative unit, as a binary feature. We only included pairs that do not contain a stopword and occurred in at least one percent of all the training instances.

5.4 NN Model Optimization and Configurations

Optimization. We split the data into a training and a test set as described in section 5.1. For development purposes, we once randomly split off 200 examples from the training data of Debatepedia and Essays and 100 examples from the smaller Microtexts data. Let the training data be defined as $D = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i consists of a source and target argument unit and $y_i \in \{0, 1\}^2$ is the one-hot vector corresponding to the two relation classes: (*support, attack*). Let, for any datum indicated by i, $p_{i,s}$ be the *support*-probability assigned by our model and $p_{i,a}$ the *attack*-probability. Using stochastic mini batch gradient descent (batch size: 32) with Adam [12], we minimize the categorical cross entropy loss over the training data, H, computed as in Equation 1:

$$H = -\frac{1}{N}\sum_{i=1}^{N}(y_{i,s} \cdot \log p_{i,s} + y_{i,a} \cdot \log p_{i,a}), \tag{1}$$

where $y_{i,s} = 1$ if observation i is classified as *support* and 0 otherwise (and similarly $y_{i,a} = 1$ if observation i is classified as *attack* and 0 otherwise). We optimize all parameters of the model except the word embeddings.

Configurations. Building on our basic Siamese model (**NN**), we inject (i), the graph features derived from ConceptNet (**NN+CN**); (ii), the same features but derived from DBpedia (**NN+DB**) and (iii), a concatenation of both (**NN+DB+CN**). For comparison purposes, we also run experiments using only the feature vector derived from the knowledge base. This is achieved by basing the classification only on this feature vector (obtained from DBpedia (**DB**), ConceptNet (**CN**) or DBpedia+ConceptNet (**DB+CN**)), ignoring and leaving out the embedded relation. Instead of concatenating knowledge features to our Siamese relation classification model, we also perform experiments where we concatenate the linguistic feature vector to the argument relation embedding (**NN+Ling**). Our full-feature argumentative relation classification model is **NN+Ling+CN+DB**.

5.5 Results

Table 2 presents the F1 scores that our evaluated models obtain on all three datasets. The main observation is that overall, the knowledge base enhanced model **NN+Ling+CN+DB** achieves the best results. Second, the baselines **Ling, random** and **majority** are outperformed by *all* configurations of the neural Siamese model **NN** on *all* three data set.

The performance of our basic Siamese model (**NN**), for almost all evaluation metrics and data sets, is situated between **Ling** and all NNs which are augmented with knowledge. **NN** outperforming **Ling** indicates that the neural model is able to capture surface features not explicitly modeled by **Ling**. However the combination **NN+Ling** does achieve better results than **NN** suggesting that the two types of features are complementary.

With respect to knowledge enhanced models, both **NN+CN** and **NN+DB** outperform **NN** in terms of macro-F1, indicating that they manage to successfully use external knowledge. However, our experiments show no benefit from bringing together features

Table 2 Results over different systems and data sets.

	F1 scores								
	Debatepedia			Microtexts			Essays		
	support	attack	macro	support	attack	macro	support	attack	macro
random	$50.2^{\pm 1}$	$50.1^{\pm 1}$	$50.2^{\pm 1}$	$73.0^{\pm 5}$	$27.8^{\pm 11}$	$50.4^{\pm 8}$	$89.2^{\pm 1}$	$10.5^{\pm 4}$	$49.8^{\pm 3}$
majority	**66.3**	0.0	33.2	82.1	0.0	41.1	**94.9**	0.0	47.5
Ling	61.4	49.8	55.6	73.3	42.9	58.1	**94.9**	0.0	47.5
DB	43.7	56.8	50.2	81.1	0.0	40.5	94.8	0.0	47.4
CN	45.6	55.1	50.3	65.9	31.8	48.9	**94.9**	0.0	47.5
DB+CN	46.4	55.3	50.8	82.1	0.0	41.1	**94.9**	0.0	47.5
NN+Ling	58.1	55.7	56.9	77.7	35.2	66.7	92.7	20.7	56.7
NN	58.6	57.6	58.1	74.2	46.5	60.3	78.7	17.1	47.9
NN+DB	56.8	**59.7**	58.2	77.4	46.2	61.8	84.1	19.5	51.8
NN+CN	60.3	56.8	**58.6**	**83.5**	41.4	62.4	86.5	20.2	53.3
NN+DB+CN	58.6	57.6	58.1	81.2	38.7	59.9	88.0	16.3	52.1
NN+Ling+CN+DB	58.6	56.2	57.4	82.5	**51.4**	**67.0**	91.2	**25.7**	**58.7**

Table 3 Number of cases which were labeled incorrectly by the **NN** baseline but correctly by another model minus the number of cases which were labeled correctly by the NN baseline but incorrectly by another model. Worst and **best** values are highlighted.

	vs. NN baseline								
	Debatepedia		Microtexts		Essays		Total		
	Δ sup.	Δ att.	Δ sup.	Δ att.	Δ sup.	Δ att.	Δ sup.	Δ att.	Δ att. + Δ sup.
Ling	**153**	-231	0	-2	**95**	-12	**248**	-245	3
NN+DB	-47	**22**	3	**-1**	26	**-1**	-18	**20**	2
NN+CN	63	-78	**10**	-4	39	-2	107	-84	**23**
NN+DB+CN	13	-43	8	-4	49	-5	70	-52	18

from both ConceptNet and DBpedia on top of the **NN** system, a result that requires more investigation. Nevertheless, when ConceptNet and DBpedia features are brought together on top of **NN+Ling** features, the system achieves the best results. Training a linear classifier solely with the background knowledge features achieves lower results than the **Ling** baseline, and also lower than all other configurations on top of **NN**. This indicates that the knowledge features are only useful when in conjunction with text based features.

With respect to the two targeted argumentative relation classes, *attack* relations are more challenging to capture in the Microtexts and Essays datasets, because of the very low frequency in the data (see Table 1). It is interesting to notice that on our biggest and most balanced dataset (Debatepedia), **NN+DB** provides more accurate detection of *attack* relations than of *support* relations, and that overall the settings that use DBpedia achieve better results at detecting the *attack* relation, than the settings that do not use DBpedia. This might be because DBpedia does not capture lexical knowledge, therefore attacking concepts lie further away in the graph than they do in ConceptNet. This is a very interesting insight and worth more investigation in the future.

Comparative Analysis of the Neural Models. To give deeper insights into the performances of our knowledge enhanced models, we present a deeper comparison between them and the **NN** and **Ling** predictions. The results over all three data sets are displayed in Table 3. In total, **NN+CN** provides most corrections of otherwise falsely classified cases (+23 over all data sets; −15 on Debatepedia, +6 on Microtext and +37 on Essays). A correction of a false-positive attack label (+107 in total) appears to be more likely than a correction of a

Table 4 Examples from Microtext and Essays which were assigned a significantly higher probability for the correct label by the knowledge-augmented model (**NN+CN**) compared to our neural baseline model (**NN**).

argumentative unit A (source)	argumentative unit B (target)	y	Δ
prohibition has kept marijuana out of children's hands	prohibition does more harm than good	ATT	0.66
using technology or advanced facilities do not make food lose its nutrition and quality	investing much time in cooking food will guarantee nutrition as well as quality of food for their family	ATT	0.15
they will have a bad result in school	even people who are not interested in online game can still be negatively affected by using computer too much	SUP	0.84
Education and training are fundamental rights which the state , the society must provide	Tuition fees should not generally be charged by universities	SUP	0.38

false-positive support label, in fact, for the attack label, the knowledge augmented model makes more mis-corrections than corrections (−84 in total, with the strongest such effect on Debatepedia). This means that the knowledge helps the model in determining support relations more than in determining attack relations. Overall, the knowledge-enhanced models, especially **NN+CN**, tend to have a better overall correction ratio compared to Ling.

Examples. To understand where the injection of background knowledge helps the most, we investigated the AU pairs which were falsely classified by **NN** but correctly classified by **NN+CN**.We rank these cases according to the margin $p_{NN+CN}(c) - p_{NN}(c)$, where $p(c)$ is the probability of the correct class. Four cases with large margins are displayed in Table 4. In the first example, there is only one explicit link in the form of a shared word (*prohibition*). The attack-relation has its foundation in the fact that A probably views prohibition (of marijuana) rather positively. His belief is based on the premise that children are protected by prohibition – the protection of children from drugs is widely considered as something highly desirable. On the other hand, B views prohibition more negatively and thus B can consider itself attacked by A. The baseline **NN** mislabeled the relation as a *support* relation, assigning the attack relation a low probability. The knowledge augmented model, in contrast, predicted the correct label very confidently. All four examples have in common that there are no shallow markers which somehow could predict the outcome. For proper resolution of these examples, knowledge about the world needs to be applied in conjunction with knowledge about syntax (e.g., by removing the negation from the fourth example, the support relation transforms into a attack relation).

6 Conclusion

In this paper, we have investigated the use of background knowledge for argumentative relation classification. We introduced a Siamese neural network system that uses word embeddings and can be enriched with specifically designed feature vectors. We designed features that exploit knowledge graphs such as ConceptNet and DBpedia and evaluate their usefulness. Experimental results on three different corpora show that knowledge based features capture aspects that are complementary to the surface features, and can substantially improve the classification results.

Our presented study is a first step towards a knowledge-rich argument analysis and opens new research directions into investigating and exploiting knowledge graphs for argumentation understanding. We plan to explore more sophisticated ways to make use of background knowledge for argumentation structure reconstruction and for explaining arguments.

References

1 Stergos Afantenos, Andreas Peldszus, and Manfred Stede. Comparing decoding mechanisms for parsing argumentative structures. *Argument and Computation*, 9:177–192, 2018.

2 Maria Becker, Michael Staniek, Vivi Nastase, and Anette Frank. Enriching Argumentative Texts with Implicit Knowledge. In Flavius Frasinca, Ashwin Ittoo, Le Minh Nguyen, and Elisabeth Metais, editors, *Applications of Natural Language to Data Bases (NLDB) - Natural Language Processing and Information Systems*, Lecture Notes in Computer Science. Springer, 2017. URL: `http://www.cl.uni-heidelberg.de/~mbecker/pdf/enriching-argumentative-texts.pdf`.

3 Teresa Botschen, Daniil Sorokin, and Iryna Gurevych. Frame- and Entity-Based Knowledge for Common-Sense Argumentative Reasoning. In *Proceedings of the 5th Workshop on Argument Mining*, pages 90–96, 2018. URL: `http://aclweb.org/anthology/W18-5211`.

4 Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

5 Zhimin Chen, Wei Song, and Lizhen Liu. TRANSRW at SemEval-2018 Task 12: Transforming Semantic Representations for Argument Reasoning Comprehension . In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1142–1145, 2018. `doi:10.18653/v1/S18-1194`.

6 HongSeok Choi and Hyunju Lee. GIST at SemEval-2018 Task 12: A network transferring inference knowledge to Argument Reasoning Comprehension task . In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 773–777, 2018. `doi:10.18653/v1/S18-1122`.

7 Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. SemEval-2018 Task 12: The Argument Reasoning Comprehension Task. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 763–772, 2018.

8 Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

9 Yufang Hou and Charles Jochim. Argument Relation Classification Using a Joint Inference Model. In *Proceedings of the 4th Workshop on Argument Mining*, pages 60–66, 2017.

10 Ioana Hulpuş, Narumol Prangnawarat, and Conor Hayes. Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *International Semantic Web Conference*, pages 442–457. Springer, 2015.

11 Taeuk Kim, Jihun Choi, and Sang-goo Lee. SNU_IDS at SemEval-2018 Task 12: Sentence Encoder with Contextualized Vectors for Argument Reasoning Comprehension . In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1083–1088, 2018. `doi:10.18653/v1/S18-1182`.

12 Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014. `arXiv:1412.6980`.

13 Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. Recognizing Implicit Discourse Relations in the Penn Discourse Treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 343–351. Association for Computational Linguistics, 2009. event-place: Singapore. URL: `http://aclweb.org/anthology/D09-1036`.

14 Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. URL: http://www.aclweb.org/anthology/P/P14/P14-5010.

15 Stefano Menini and Sara Tonelli. Agreement and Disagreement: Comparison of Points of View in the Political Domain. In *COLING*, pages 2461–2470, 2016.

16 Huy Ngoc Nguyen and Diane J. Litman. Context-aware Argumentative Relation Mining. In *ACL*, pages 1127–1137, 2016.

17 Andreas Peldszus and Manfred Stede. From Argument Diagrams to Argumentation Mining in Texts: A Survey. *Int. J. Cogn. Inform. Nat. Intell.*, 7(1):1–31, January 2013. doi:10.4018/jcini.2013010101.

18 Andreas Peldszus and Manfred Stede. Joint prediction in MST-style discourse parsing for argumentation mining. In *EMNLP*, pages 938–948, 2015.

19 Andreas Peldszus and Manfred Stede. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon 2015 / Vol. 2*, pages 801–815, London, 2016. College Publications.

20 Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

21 Isaac Persing and Vincent Ng. End-to-End Argumentation Mining in Student Essays. In *HLT-NAACL*, pages 1384–1394, 2016.

22 Peter Potash, Robin Bhattacharya, and Anna Rumshisky. Length, Interchangeability, and External Knowledge: Observations from Predicting Argument Convincingness. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 342–351, 2017.

23 Push Singh. The Open Mind Common Sense Project, 2002. URL: http://zoo.cs.yale.edu/classes/cs671/12f/12f-papers/singh-omcs-project.pdf.

24 Robert Speer and Catherine Havasi. Representing General Relational Knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3679–3686, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

25 Christian Stab and Iryna Gurevych. Annotating Argument Components and Relations in Persuasive Essays. In *COLING*, pages 1501–1510, 2014.

26 Christian Stab and Iryna Gurevych. Identifying Argumentative Discourse Structures in Persuasive Essays. In *EMNLP*, pages 46–56, 2014.

27 Christian Stab and Iryna Gurevych. Parsing Argumentation Structures in Persuasive Essays. *Computational Linguistics*, 43:619–659, 2017.

Graph-Based Annotation Engineering: Towards a Gold Corpus for Role and Reference Grammar

Christian Chiarcos
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
http://acoli.informatik.uni-frankfurt.de/
chiarcos@informatik.uni-frankfurt.de

Christian Fäth
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
faeth@em.uni-frankfurt.de

Abstract

This paper describes the application of annotation engineering techniques for the construction of a corpus for Role and Reference Grammar (RRG).

RRG is a semantics-oriented formalism for natural language syntax popular in comparative linguistics and linguistic typology, and predominantly applied for the description of non-European languages which are less-resourced in terms of natural language processing. Because of its cross-linguistic applicability and its conjoint treatment of syntax and semantics, RRG also represents a promising framework for research challenges within natural language processing. At the moment, however, these have not been explored as no RRG corpus data is publicly available. While RRG annotations cannot be easily derived from any single treebank in existence, we suggest that they can be reliably inferred *from the intersection* of syntactic and semantic annotations as represented by, for example, the Universal Dependencies (UD) and PropBank (PB), and we demonstrate this for the English Web Treebank, a 250,000 token corpus of various genres of English internet text. The resulting corpus is a gold corpus for future experiments in natural language processing in the sense that it is built on existing annotations which have been created manually.

A technical challenge in this context is to align UD and PB annotations, to integrate them in a coherent manner, and to distribute and to combine their information on RRG constituent and operator projections. For this purpose, we describe a framework for flexible and scalable annotation engineering based on flexible, unconstrained graph transformations of sentence graphs by means of SPARQL Update.

2012 ACM Subject Classification Computing methodologies → Language resources; Information systems → Semantic web description languages; Computing methodologies → Natural language processing; Computing methodologies → Lexical semantics

Keywords and phrases Role and Reference Grammar, NLP, Corpus, Semantic Web, LLOD, Syntax, Semantics

Digital Object Identifier 10.4230/OASIcs.LDK.2019.9

Category Short Paper

Supplement Material The software described in this paper are available under the Apache 2.0 license from https://github.com/acoli-repo/RRG. This includes build scripts for the data. We aim to provide the data under the same license as the annotations it is derived from (CC-BY-SA), but we are still in the process of copyright clearance for the original text.

Funding The research described in this paper was conducted in the context of the project *Linked Open Dictionaries* (LiODi, 2015-2020), funded by the German Ministry for Education and Research (BMBF), as well as the project *Specialised Information Service Linguistics* (Fachinformationsdienst Linguistik, funding period 2017-2019) funded by the German Research Foundation (DFG).

© Christian Chiarcos and Christian Fäth;
licensed under Creative Commons License CC-BY

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 9; pp. 9:1–9:11

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Annotation engineering can be defined as the task to transform linguistic annotations from one or several specific source representations into representations of a different type or quality in an automated or semi-automated fashion.

The goal of annotation engineering is to produce high-quality annotations (gold data) for training state of the art tools in natural language processing. It is thus an essential aspect for the growth of the discipline beyond earlier annotation efforts, as it allows legacy resources created with great manual effort to be re-used as test and training data for a novel theoretical framework. The goal of annotation engineering is *not* to replace annotation tools, but rather to provide training data for them. As it posits particularly high standards of annotation quality, annotation engineering is to be done in a transparent, rule-based fashion rather than by machine learning. The outcome of annotation engineering is a resource, i.e., an annotated corpus or information extracted from it, so that transformation efficiency plays a considerably less important role than conversion quality. Typical examples for annotation engineering include, for example, transformations of annotations between different theoretical frameworks, e.g., and this will be illustrated here for the case of Role and Reference Grammar.

Annotation engineering differs from plain conversion in the sense that its output is typically qualitatively different or richer than the source annotations. This can be achieved, for example, by including additional knowledge sources, e.g., lexical resources, or additional sources of annotation. To a large extent, such resources are already available from the web of data, where specifications for the publication of (open) language resources are developed in the context of the Linguistic Linked Open Data (LLOD) cloud and associated W3C community and business groups.[1] In the last decade, a large number of language resources have been published in this context, in accordance with W3C standards and as Linked Data, and with their formal metadata registered at portals such as LingHub,[2] thus facilitating their usability and interoperability. The Linguistic Linked Open Data Cloud comprises corpora, dictionaries, resource metadata and terminology repositories and knowledge bases which are made interoperable through the use of shared vocabularies and ontologies such as GOLD, ISOcat, OLiA, lexvo and lexinfo. As of January 2019, over 100,000 resources covering more than 1,000 languages are listed on LingHub, a curated subset of open resources of these is the basis for the LLOD cloud diagram.

In order to facilitate the integration of such resources in annotation engineering workflows, with CoNLL-RDF [5], we developed an approach based on LOD standards, most notably RDF for representing annotations, and SPARQL Update[2] for their transformation. In conceptual terms, these allow to render, manipulate and create arbitrary linguistic annotations in the form of labeled directed multi-graphs – and, as established already by Bird and Liberman [1] –, they are thus capable of encoding *every* kind of linguistic annotation.

In this paper, we illustrate the application of our annotation engineering approach on the creation of a Role and Reference Grammar treebank. Aside from the conceptual challenge to derive a gold corpus from existing manual annotations, a technical challenge in this context is that RRG syntax cannot be derived from any common treebank formalism, but instead, it requires to create and to process the intersection of independent annotations for syntax and semantics.

[1] `http://linguistic-lod.org/llod-cloud`
[2] `http://linghub.org/`

2 Role and Reference Grammar

Role and Reference Grammar [8, RRG] is a theory of grammar developed by Robert D. Van Valin, Jr. and William A. Foley during their research of Austro-Asiatic and native American languages. Encountering various problems in applying established theories of grammar, they aimed to overcome the European bias in language description by devising a descriptive grammar formalism that integrates structural analyses with semantics and pragmatics.

RRG features several *projections*, i.e., interdependent levels of analyses all grounded in the surface expression of a particular sentence. The RRG *constituent projection* is a phrase structure representation of content elements (excluding operators), complemented by the *operator projection* that formalizes scope and attachment of function words, and the *focus projection* that identifies speech acts and information structures. Semantics has a different status, as (the constituent projection in) RRG syntax is designed to reflect frame semantics. In fact, argument linking has been a priority in the design of RRG: RRG formalizes a linking algorithm that connects semantic (macro) roles (ACTOR and UNDERGOER) in an underlying lexical inventory with the surface grammar and vice versa: The CORE of an utterance contains the semantic predicate and its core arguments, the PERIPHERY contains its (semantic) modifiers. Every CORE thus corresponds exactly to an instance of a verbal frame in the sense of FrameNet.[3] In this paper, we focus on constituent projection and operator projections in RRG as illustrated in Fig. 2.[4]

For reasons of space, we cannot explain the specifics of RRG here in detail. Yet, one aspect that is worth mentioning is that RRG postulates a direct relationship between functional operators and the nodes they modify, e.g., aspect applies to the semantic nucleus (NUC) of a clause, modality applies to its CORE, and tense to the CLAUSE itself. This has implications for juncture and nexus: If two verbs are connected by a paratactic relation in surface syntax, the scope of shared operators (and shared arguments) define whether this is a co(sub)ordination at the level of NUC, CORE, CLAUSE or SENTENCE. As a result, RRG constituent projections cannot be derived from any other syntax formalism but need to be adjusted to account for theory-specific constraints arising from the occurrence of function words and/or overlap in semantic arguments. While this raises problems for bootstrapping RRG syntax from existing annotations, it is an advantage in terms of expressivity, as RRG structure is designed to provide an appropriate representation of both frame semantics and operator scope.

Because of its semantics-based approach to syntax, RRG poses a promising framework for developing innovative applications in natural language understanding processing and natural language understanding. Furthermore, RRG ties in with the current trend towards intensified research of less-resource languages, as indeed, RRG is a popular theoretical framework for language documentation and linguistic typology. If RRG can be operationalized for one language, say, English, this paves the way for improved NLP support for languages to which RRG has been previously applied, e.g., languages from the Americas (Yaqui, Zoque, Lakota, etc.), the Pacific (Amele, Kankanaey, Amis, etc.), Asia (Zaza, Farsi, Tibetan, etc.), and to

[3] See `http://framenet.icsi.berkeley.edu/`. As for the relation between CORE and higher elements of syntactic analysis such as recursive CORES, CLAUSE, SENTENCE and *TEXT*, these are created for the purpose of juncture, nexus and extraposition, but all require a non-recursive CORE as their basis.

[4] This visualization has been created with TIGER Search [10] and follows its graphical design: Primary edges (black) represent the constituent projection, secondary edges (green) represent the operator projection, the target nodes of secondary edges would be shared in both projections.

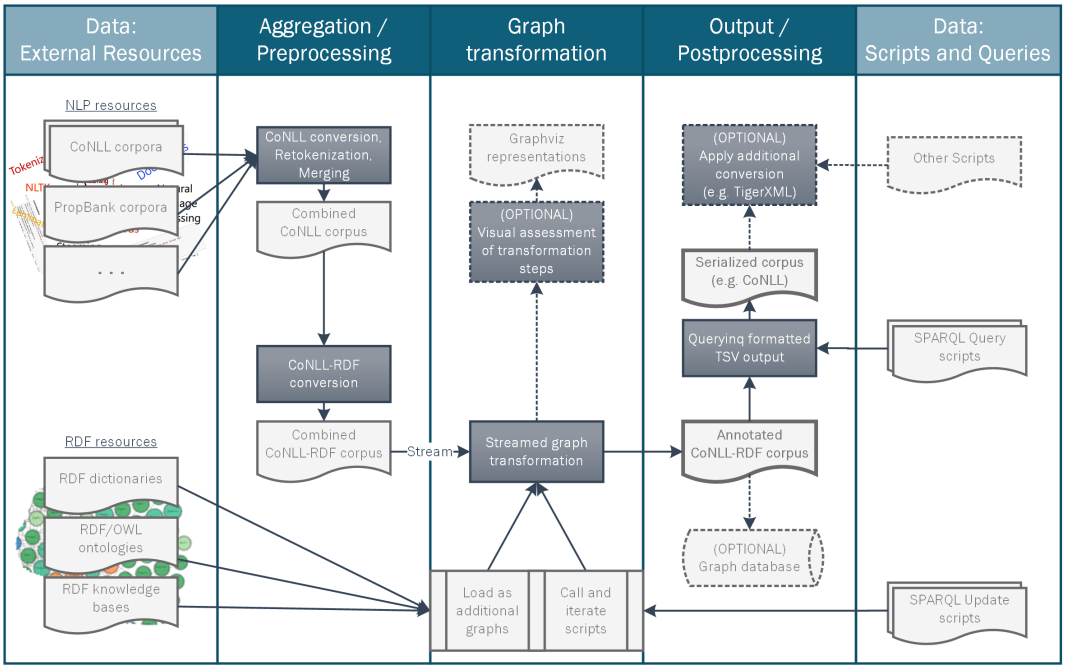


Figure 1 Annotation Engineering Workflow.

some extent Africa (Hausa, Gikuyu).

A fundamental problem in the practical application of state-of-the-art NLP techniques to RRG is, however, that no annotated data is currently available in order to assess potential challenges and benefits RRG may pose to state-of-the-art NLP techniques based on Machine Learning. Here, we address this deficit by creating an English RRG Treebank from manual annotations of syntax and frame semantics by means of annotation engineering.

3 Graph-based Annotation Engineering

The goal of annotation engineering is to produce high-quality annotations (gold data) for training and evaluating NLP tools. In our case, we ground RRG clausal syntax in existing manual annotations for semantic frames (PropBank, [15, PB]), and derive remaining aspects of constituent and operator projection from independent manual annotations for dependency syntax and morphosyntax provided as part of the Universal Dependencies [14, UD] corpora. PropBank and UD overlap in the English Web Treebank (EWT), a corpus of 250,000 words representing various genres of English internet text, and we adopt the English Web Treebank [17, EWT] as the basis for bootstrapping an RRG Treebank for English.

However, these formats bear structural limits which have to be worked around. This applies especially for back-reference in natural language resulting in separated phrases sharing the same head, thus violating the purely hierarchical tree structure. While this can be avoided by restructuring the tree or inserting NULL tokens (as in PTB) it produces challenges in data modeling which could easily be avoided by implementing a purely graph-based format which is able to host and integrate all data structures at the same time using labeled edges.

3.1 Annotation engineering with the ACoLi CoNLL libraries

The AColi CoNLL libraries [7] provide native support for the CoNLL-U format as provided by UD, and they support specific vocabulary extensions for semantic frames (SRL annotations) in CoNLL TSV formats, they are thus adopted here as the technical basis for annotation engineering. Building on that, we provide a generic, customizable workflow (Fig. 1) for the integration of heterogeneously annotated corpora into a common data structure and the induction of novel data structures. We employ SPARQL Update rules to perform advanced graph transformations, and the CoNLL-RDF API[5] supports both their aggregation and organization in files, as well as their sequential iteration.

CoNLL-RDF supports various output formats, most importantly tabular data in TSV or CoNLL, RDF in different serializations, a "canonical" Turtle representation that emulates the appearance of CoNLL-TSV formats, human-readable representations, and Dot/GraphViz plots. As exchange format for the RRG corpus, we provide its data (without diagnostic information) as TIGER/XML [11] using a specialized export functionality [6].

3.2 Integration of concurrent annotations

Resource integration can be performed on multiple levels. A particularly challenging aspect of annotation engineering is the combination of concurrent, independently performed annotations of the same text, by different groups using different tools and formats. Using converters bundled with the ACoLi CoNLL libraries or available from third parties, most annotation formats can be converted to a CoNLL format or a corresponding TSV representation. CoNLL-RDF supports the flexible handling of differences with respect to the order and naming of columns. Conflicting tokenizations (i.e., differences in the definition of rows) are handled by CoNLL Merge.[6] We merged UD files with the corresponding PropBank files via their shared part-of-speech columns (the PB skel files do not include the text), successfully merging 99.99% of the EWT tokens (254,564 of 254,593) in default mode. The remaining 29 mis-aligned tokens were manually corrected.[7] Merging resulted in a single TSV file holding all information from UD and PB, ideally suited for subsequent conjoint processing.

3.3 Graph creation

CoNLL-RDF allows us to render tabular data structures from CoNLL or related one-word-per-line TSV formats as an RDF graph. The `CoNLLStreamExtractor` retains the order of tokens, columns and sentences within a corpus but adds minimal additional overhead to make it RDF-compliant. The idea of the converter is to identify words by URIs, to define them as `nif:Word`s and as being connected by `nif:nextWord`, and to use user-provided labels (say, `WORD`) to map every column to a property of the same name in the `conll:` namespace. Listing 1 is an example fragment for the analysis of the sentence *Where can I get morcillas in Tampa Bay* (EWT, dev/answers/20070404104007AAY1Chs_ans).

[5] `https://github.com/acoli-repo/conll-rdf`
[6] `https://github.com/acoli-repo/conll`
[7] CoNLL Merge also provides a *force* mode that enforces the tokenization of one file. Because of the minimal number of misalignments, however, this was not necessary here.

Listing 1 CoNLL-RDF canonical format.

```
:s1_1 a nif:Word; conll:WORD "where"; conll:HEAD :s1_4; ... ; nif:nextWord s1_2 .
...
:s1_4 a nif:Word; conll:WORD "get"; conll:HEAD :s1_0; ... ; nif:nextWord s1_5 .
...
:s1_6 a nif:Word; conll:WORD "in"; conll:HEAD :s1_4; ... ; nif:nextWord s1_7 .
:s1_7 a nif:Word; conll:WORD "tampa"; conll:HEAD :s1_8; ... ; nif:nextWord s1_8 .
:s1_8 a nif:Word; conll:WORD "bay"; conll:HEAD :s1_6; ... .
```

3.4 Graph transformation

The `CoNLLRDFUpdater` is designed for the stream processing of large corpora that cannot be held in memory: It reads a single sentence, creates an RDF graph for it, optionally adds context information (e.g., from preceding context or from a background knowledge graph loaded at initialization), applies SPARQL update transformations and spells out the results. Sentence-by-sentence processing minimizes memory usage and search space for the transformations, and in addition, the process is parallelized to improve run-time performance.

When processing corpora, we read the input from stdin, apply the `CoNLLStreamExtractor` to produce RDF graphs and the `CoNLLRDFUpdater` for graph transformation. Graph transformations are implemented with SPARQL Update rules, in that a sequence of SPARQL files can be provided as arguments for the `CoNLLRDFUpdater`. Each file contains a number of SPARQL update operations which are executed in their sequential order. In addition, numerical flags allow each file to be executed multiple times or until no further changes occur. The results are flushed through `stdout` while next sentence is read. Lookahead and lookback parameters allow cross-sentence analyses (e.g. of text coherence) while parallelization speeds up the process. Furthermore, native LOD and RDF resources can be easily integrated by either federated queries or preloading RDF dumps into separate named graphs making them available within the update scripts.

In annotation engineering and rule-based parsing continuous revision and reorganization of rules is a crucial aspect. By separating transformation operations (SPARQL) from the actual code (CoNLL-RDF classes, JAVA), the transformations become easily portable between the CoNLL-RDF environment and off-the-shelf triple stores as well as between different workflows that require the same functionality. SPARQL statements can thus be run independently, e.g., on a database snapshot and easily optimized on that basis. Additionally, this architecture improves reusability of modular scripts and encourages contribution in community projects.

4 Use Case: Building an RRG Treebank

Using our annotation engineering workflow, we implemented a rule-based conversion routine for transforming the UD and PropBank representation of the EWT corpora into an RRG representation.

In addition to EWT data, we also digitized and converted the complete body of English examples (429 examples, 3,766 tokens) drawn from [18, 19], using UD v.1-compliant annotations produced by the Stanford parser [3, manually corrected].[8] As these are lacking

[8] With respect to RRG annotations, we primarily follow the notational conventions of [18], in that we use the labels `ARG` and `NP`. This is upward-compatible, though: In more recent editions, the `ARG` would be just omitted (as it can be inferred from the underlying lexical inventory – which does not exist in our

semantic role annotations, heuristic rules have been devised to identify verbal predicates and their semantic arguments from the syntactic annotation alone. This extrapolation does not replace full-fledged PropBank annotations, so, where it failed, we provide explicit patterns for specific verbs and their grammatical roles as part of the transformation workflow in order to reproduce textbook examples (cf. Fig. 2.)

For representing RRG data structures, we define an RDF vocabulary for representing constituency projection and operator projection:[9] All RRG node types from the constituency projection are defined by concepts such as `rrg:NUC`, `rrg:CORE`, etc. The relations between them are formalized by two navigational properties, `rrg:has` (pointing from parent node to children), and `rrg:next` (connecting RRG nodes with the next following siblings).

The operator projection is represented by RDF properties such as `rrg:TNS` (tense) or `rrg:IF` (illocutionary force) which point from the words or phrases that evoke a particular operator to the RRG node that they are associated with (e.g., a `rrg:CLAUSE`). Listing 2 shows a (slightly simplified) example rule that creates a `rrg:NUC` from existing SRL annotations[10]

Listing 2 Graph transformation with SPARQL.

```
INSERT { _nuc a rrg:NUC; rrg:has ?pred. }
WHERE {  ?pred conll:SRL []. };
```

This rule reads the `SRL` column of the PropBank skel files that holds the disambiguated identifier of the semantic predicate (normally preceding the argument columns) and creates a syntactic nucleus if one is encountered.

4.1 Transformation steps

The transformation workflow consists of the following steps, each corresponding to a SPARQL Update file. The actual implementation of this workflow in CoNLL-RDF merely requires to provide this list of SPARQL Update files to the `CoNLLRDFUpdater` after `-update`:

constituents. Identify noun phrases, prepositional phrases, modifier (adverb, adjective) phrases.

clausal structure. For every (verbal) PB predicate, create clausal NUC, CORE, CLAUSE, SENTENCE together with arguments, periphery, PrCS and dislocation positions. Note that the resulting data structure is not a tree: Multiple COREs can share their arguments, etc.

operators. For every word, identify all NUC- (CORE-, CLAUSE-) level operators and to the NUC (CORE, CLAUSE) that comprises the UD head of the word. It is important here that this builds on the operator hierarchy specified by RRG as it guides constituent pruning during juncture assessment.

juncture and nexus. This is the most challenging and RRG-specific part of the conversion process:

1. Constituent pruning: Eliminate all SENTENCE (CLAUSE, CORE) nodes that feature neither a operator nor an argument.

---

case), and `NP` would be renamed to `RP`. Furthermore, we adopt a simplified handling of `ADJ` and `ADV` phrases for which a novel `MP` node has been introduced at a later point in time.

[9] `PREFIX rrg: <http://www.acsu.buffalo.edu/~rrgpage/rrg.html#syn_>`

[10] For identifying the newly created NUC element, this listing uses a blank node. In the actual implementation, we provide full URIs.

2. Co(sub)ordination: If a UD `conj` (similar for `ccomp`, `xcomp`, `parataxis`, etc.) holds between two unconnected NUCs (COREs, CLAUSEs), create NUC (etc.) coordination under the same CORE (CLAUSE, SENTENCE) parent. If two coordinated COREs (CLAUSEs) share an argument *in the same semantic role* and use the same (or no) CORE- (CLAUSE-) level operators, transform the coordination into CORE (CLAUSE) cosubordination by inserting a CORE (CLAUSE) node that contains both COREs (CLAUSEs).
3. Establish tree structures by eliminating shared arguments.
4. Subordination: If an UD `acl` (similar for `advcl`, etc.) holds between two unconnected tree fragments, attach the tree containing the dependent to the smallest suitable constituent that contains the UD head.
5. Sentence completion: For every tree fragment that does not contain a SENTENCE, create its RRG parent nodes up to the level of SENTENCE. Connect multiple SENTENCE nodes within an utterance by a TEXT node.

4.2 Evaluation

Beyond the textbook examples mentioned above, we are not aware of any source for RRG gold annotations for English, so that we currently have no basis for a quantitative extrinsic evaluation. Instead, we performed an intrinsic evaluation by means of two optional validation steps, provided in two SPARQL files:

Structural validation. Establishes tree structures. If a node has more than two parents, it will be disconnected and assigned a new `MPARENT` node as parent. If an utterance contains more than one partial tree, these will be joined under a new `FRAG` node. This includes any unattached word, but exceptions apply (e.g., RRG does not account for punctuation).

Pattern validation. The original RRG parsing algorithm defines parsing templates. As these have not been used for creating the parses, we adopt them for validating parses. Here, we implement each of these templates as WHERE conditions in SPARQL INSERT statements, with diagnostic information (e.g., provenance of this particular pattern in the textbook) inserted as an `rdfs:isDefinedBy` reference. The number of templates is relatively high, so that pattern validation is slow (and optional).

The RRG generation workflow was exclusively modeled on RRG text book examples and the EWT development set from the answers domain. We focused on answers as this portion is particularly challenging in that it often contains grammatical errors. This may include, for example, omissions of function words, insertion of incorrect function words when written by non-natives or in a careless fashion. Like most linguistic phenomena, errors, and annotation gaps follow a distribution with a long tail, we thus do not guarantee convertability for the entire EWT data, but we aimed for an RRG-compliant conversion of 90% of the corpus to provide a suitable starting point for developing NLP tools. For the remaining (up to) 10%, subsequent efforts for manual correction of this data are necessary.

Ultimately we were able to convert 98.4% (418/425) of dev/answers sentences into structurally valid RRG representations, resp. 92.6% (1828/1974) sentences from the EWT development set in general, and 91.2% (1880/2061) from the EWT test set.

We also performed pattern validation, with 74.9% (1479/1974) and 73.9% (1524/2061) structurally and pattern-valid sentences on development set and test set, respectively. It should be noted, however, that we only validated against patterns as explicitly found in the text books or necessary for text book examples, so that low number for pattern validation may less reflect invalid RRG parses than they reflect gaps in the pattern inventory.

Along with this publication, we prepare the release of the corpus under `https://github.com/acoli-repo/RRG`, with the aim to reach out to the RRG community to elicit feedback and bug reports as a source of extrinsic (qualitative) evaluation and further improvement.

5 Discussion

Graph-based or graph-assisted parsing has had a long history in natural language processing. In the late 1980s [16] proposed the usage of graphs instead of trees for representing syntactic structures. Over the past decades, new approaches emerged resulting in growing performance improvements.[9] showed how statistical parsers could be augmented by graph transformation. [13] gained similar improvements with post processing the results of data-driven dependency parsers. In semantics, knowledge bases are mostly represented as graphs leading to numerous parsing approaches, e.g. using staged query graph generation [20]. Since history on this field is very diverse, it is impossible to list all relevant publications. For further reading we recommend the overview provided by [12].

Neither of these graph-based approaches, however, does address the specific task pursued here, i.e., the combination of existing annotated corpora and their transformation into a completely new representation formalism in order to alleviate the generation of gold data. Instead, they try to improve existing annotations mostly by training neural networks to efficiently stack mathematical algorithms and general transformation rules. The goal of annotation engineering, however, is *not* to replace (nor to improve) annotation tools, but rather to provide training data for them. As it posits particularly high standards of annotation quality, annotation engineering is to be done in a transparent, rule-based fashion rather than by machine learning.

In both regards, high demand for quality and the need of human supervision, annotation engineering is comparable to traditional grammar engineering (i.e., rule-based parsing). On the one hand, it is a simpler task in the sense that it transforms existing annotations rather to create them from scratch, and in particular, it does not depend on the conjoint development of lexical resources along with the rules. In particular, annotation engineering differs from grammar engineering as it is concerned with the analysis of a *finite* set of symbols rather than with the analysis of an *infinite* set of symbols from a finite set of categories.

On the other hand, the data structures encountered in annotation engineering can be *much* more diverse than those encountered in grammar engineering: In symbolic parsing, input data is a plain sequence of tokens, whereas internal and output structures are implementation-specific. In annotation engineering, input data can carry *any* kind of annotation originating from multiple sources. In comparison to grammar engineering, unrestricted annotation engineering is thus less challenging in terms of coverage, but more challenging in terms of diversity.

This paper showed that graph-based annotation engineering does have a place in NLP and that existing technologies developed in the context of the Linked Open Data community can be applied for the purpose. It does not replace machine learning, but rather serves as a technique for generating gold data for underresourced languages or annotation schemes.

Upon copyright clearance for the contained text, we will provide the corpus in three editions: Development edition (using the internal RDF vocabulary), an OWL release version in RDF, using the POWLA [4] vocabulary for generic linguistic annotations, and an release version in TIGER/XML [11] for further processing in conventional corpus tools.

References

1 Steven Bird and Mark Liberman. Annotation Graphs as a Framework for Multidimensional Linguistic Data Analysis. In Marilyn Walker, editor, *Towards Standards and Tools for Discourse Tagging*, Maryland, USA, June 1999. Association for Computational Linguistics. URL: http://aclweb.org/anthology/W99-0301.

2 Carlos Buil Aranda, Olivier Corby, Souripriya Das, Lee Feigenbaum, Paula Gearon, Birte Glimm, Steve Harris, Sandro Hawke, Ivan Herman, Nicholas Humfrey, Nico Michaelis, Chimezie Ogbuji, Matthew Perry, Alexandre Passant, Axel Polleres, Eric Prud'hommeaux, Andy Seaborne, and Gregory Todd Williams. SPARQL 1.1 Overview. https://www.w3.org/TR/sparql11-overview, 2013.

3 Danqi Chen and Christopher D Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), Doha, Qatar, 2014. Association for Computational Linguistics (ACL)*, 2014.

4 Christian Chiarcos. POWLA: Modeling Linguistic Corpora in OWL/DL. In Elena Simperl, Philipp Cimiano, Axel Polleres, Oscar Corcho, and Valentina Presutti, editors, *The Semantic Web: Research and Applications*, pages 225–239, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

5 Christian Chiarcos and Christian Fäth. CoNLL-RDF: Linked Corpora Done in an NLP-Friendly Way. In *Language, Data, and Knowledge - First International Conference, LDK 2017, Galway, Ireland, June 19-20, 2017, Proceedings*, pages 74–88, 2017. doi:10.1007/978-3-319-59888-8_6.

6 Christian Chiarcos, Benjamin Kosmehl, Christian Fäth, and Maria Sukhareva. Analyzing Middle High German Syntax with RDF and SPARQL. In *In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 2018. European Language Resources Association (ELRA)*, 2018. URL: http://www.lrec-conf.org/lrec2018.

7 Christian Chiarcos and Niko Schenk. The ACoLi CoNLL Libraries: Beyond Tab-Separated Values. In *In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 2018. European Language Resources Association (ELRA)*, May 2018.

8 William A. Foley and Jr. Robert D. Van Valin. Role and Reference Grammar. In E.A. Moravscik and J.A. Wirth, editors, *Current approaches to syntax*, pages 329–352. Academic Press, New York, 1980.

9 Dan Klein and Christopher D. Manning. Accurate Unlexicalized Parsing. In *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003), Sapporo, Japan, 2003. Association for Computational Linguistics (ACL)*, 2003.

10 Wolfgang Lezius. TigerSearch – Ein Suchwerkzeug für Baumbanken. In *Proceedings of the 6. Konferenz zur Verarbeitung natürlicher Sprache (6th Conference on Natural Language Processing, KONVENS 2002), Saarbrücken, Germany*, 2002.

11 Andreas Mengel and Wolfgang Lezius. An XML-based Representation Format for Syntactically Annotated Corpora. In *In Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000), Athens, Greece, 2000. European Language Resources Association (ELRA)*, 2000.

12 Vivi Nastase, Rada Mihalcea, and Dragomir R. Radev. A survey of graphs in natural language processing. *Natural Language Engineering*, 21(5):665?698, 2015. doi:10.1017/S1351324915000340.

13 Jens Nilsson, Joakim Nivre, and Johan Hall. Graph Transformations in Data-Driven Dependency Parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, January 2006. doi:10.3115/1220175.1220208.

14 Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut

Tsarfaty, and Daniel Zeman. Universal Dependencies v1: A Multilingual Treebank Collection. In *In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Solvenia, 2016. European Language Resources Association (ELRA)*, May 2016.

15 Tim O'Gorman, Sameer Pradhan, Martha Palmer, Julia Bonn, Kathryn Conger, and James Gung. The New Propbank: Aligning Propbank with AMR through POS Unification. In *In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 2018. European Language Resources Association (ELRA)*, 2018.

16 Jungyun Seo and Robert F. Simmons. Syntactic Graphs: A Representation for the Union of All Ambiguous Parse Trees. *Computational Linguistics*, 15(1):19–32, March 1989. URL: `http://dl.acm.org/citation.cfm?id=68960.68962`.

17 Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. A Gold Standard Dependency Corpus for English. In *In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, 2014. European Language Resources Association (ELRA)*, 2014.

18 Robert D Van Valin, Robert D van Valin Jr, and Randy J LaPolla. *Syntax: Structure, meaning, and function*. Cambridge University Press, 1997.

19 Robert D Van Valin Jr. *Exploring the syntax-semantics interface*. Cambridge University Press, 2005.

20 Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. In *In Proceedings of the Joint Conference of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (AFNLP), Beijing, China, 2015. Association for Computational Linguistics (ACL)*, pages 1321–1331, January 2015. `doi:10.3115/v1/P15-1128`.

A RRG Example

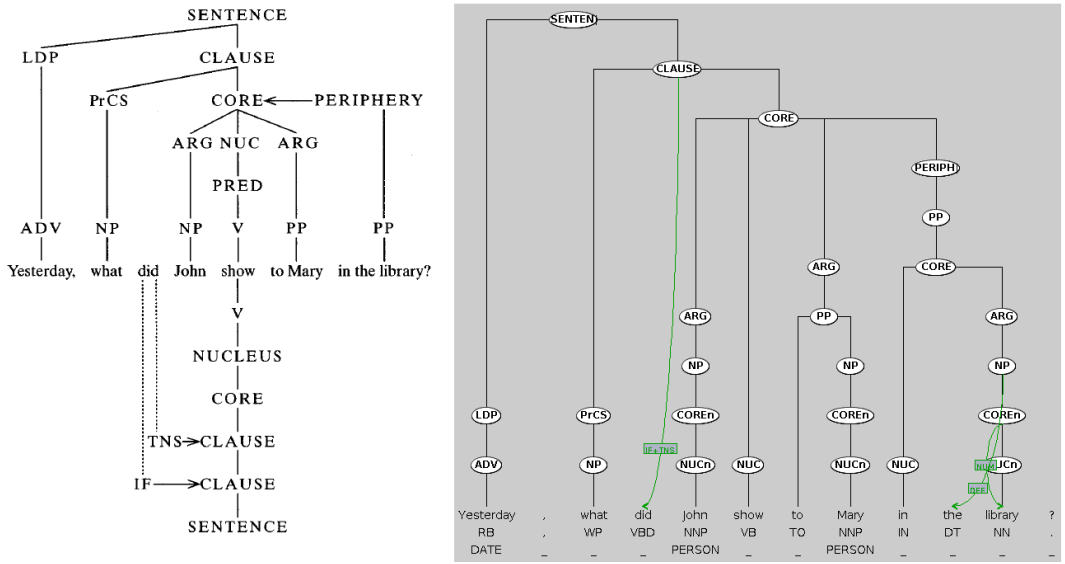


Figure 2 RRG Textbook Example(left, [18, p. 50]) compared to Synpathy rendering (right).

Crowd-Sourcing A High-Quality Dataset for Metaphor Identification in Tweets

Omnia Zayed
Insight Centre for Data Analytics, Data Science Institute, National University of Ireland Galway, IDA Business Park, Lower Dangan, Galway, Ireland
omnia.zayed@insight-centre.org

John P. McCrae
Insight Centre for Data Analytics, Data Science Institute, National University of Ireland Galway, IDA Business Park, Lower Dangan, Galway, Ireland
john.mccrae@insight-centre.org

Paul Buitelaar
Insight Centre for Data Analytics, Data Science Institute, National University of Ireland Galway, IDA Business Park, Lower Dangan, Galway, Ireland
paul.buitelaar@insight-centre.org

Abstract

Metaphor is one of the most important elements of human communication, especially in informal settings such as social media. There have been a number of datasets created for metaphor identification, however, this task has proven difficult due to the nebulous nature of metaphoricity. In this paper, we present a crowd-sourcing approach for the creation of a dataset for metaphor identification, that is able to rapidly achieve large coverage over the different usages of metaphor in a given corpus while maintaining high accuracy. We validate this methodology by creating a set of 2,500 manually annotated tweets in English, for which we achieve inter-annotator agreement scores over 0.8, which is higher than other reported results that did not limit the task. This methodology is based on the use of an existing classifier for metaphor in order to assist in the identification and the selection of the examples for annotation, in a way that reduces the cognitive load for annotators and enables quick and accurate annotation. We selected a corpus of both general language tweets and political tweets relating to Brexit and we compare the resulting corpus on these two domains. As a result of this work, we have published the first dataset of tweets annotated for metaphors, which we believe will be invaluable for the development, training and evaluation of approaches for metaphor identification in tweets.

2012 ACM Subject Classification Computing methodologies → Natural language processing; Computing methodologies → Language resources

Keywords and phrases metaphor, identification, tweets, dataset, annotation, crowd-sourcing

Digital Object Identifier 10.4230/OASIcs.LDK.2019.10

Funding This work was supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289 (Insight).

1 Introduction

Metaphor is an essential element of human cognition which is often used to express ideas and emotions. It is considered as an analogy between two concepts by exploiting common similarities. The sense of a concept such as *"war"* can be transferred to another concept's sense such as *"illness"* by exploiting the properties of the first concept. This then can be expressed in our everyday language in terms of linguistic (conventional) metaphors such as *"attack cancer"* or *"beat the illness"* [11, 17]. Among the main challenges of the computational modelling of metaphors is their pervasiveness in language which means they do not only occur frequently in our everyday language but they are also often conventionalised to such


© Omnia Zayed, John P. McCrae, and Paul Buitelaar;
licensed under Creative Commons License CC-BY

an extent that they exhibit no defined patterns. This has meant that achieving consistent annotations with higher inter-annotator agreement has been difficult and as such previous work has introduced restrictions, such as limiting the study to only a few chosen words of a certain syntactic type [1, 16, 32] or particular predefined metaphors [15, 31].

The widespread nature of Twitter communication has led to a growing interest in studying metaphors in such a context. People tend to use colloquial language in order to communicate on social media, and they may utilise figurative and metaphoric expressions more frequently. Twitter, which is the most popular microblogging application in the world, presents a new type of social media content, where users can express themselves through a tweet of limited characters. Processing metaphoric expressions in tweets can be very useful in many social media analysis applications such as political discourse analysis [3] and health communication analysis. Therefore, our goal is to create a dataset of tweets annotated for metaphors that offers comprehensive coverage of metaphoric usages as well as text genre. In order to achieve that, we need to design an annotation methodology that guarantees high annotator agreement at a large scale. Accordingly, the resulting dataset can be used for the development and evaluation of metaphor processing approaches in tweets.

There are different factors that affect the creation of datasets annotated for metaphor and their annotation scheme. Among these factors are the level of metaphor analysis and the type of metaphor, in addition to the task definition and the targeted application. Examples of metaphor types include conceptual, linguistic (conventional and novel) and extended metaphors. There exist different levels of metaphoric analysis of linguistic metaphors either on the word-level (token-level) or on the phrase-level. Processing metaphors on the word-level means looking at each word in a sentence and deciding whether it is used metaphorically or not given the context, while phrase-level processing looks at pairs of words such as verb-noun or adjective-noun pairs and check the metaphoricity of the verb or the adjective given its association with the noun. Various research has been done to address both levels of processing[1]. The majority of previous approaches pertaining to metaphor identification have focused on formal well-structured text selected from a specific corpus to create datasets to model and evaluate their approaches. A common issue of all the available datasets is that they are specifically designed for a certain task definition focusing on a certain level of metaphor analysis which makes their annotation scheme difficult to generalise. Additionally, the majority of available datasets lack coverage of metaphors and text genres as they rely on predefined examples of metaphors from a specific domain during the creation process.

In this work, we introduce the first high-quality dataset annotated for phrase-level metaphor in English tweets. We propose a crowd-sourcing approach to create this dataset which is designed to ensure the dataset balance, coverage as well as high accuracy. Our approach employs an existing metaphor identification system to facilitate quick and accurate annotations as well as maintaining consistency among the annotators. We will outline the identification system used as well as the data sources in section 3 below. In this paper, we present our annotation methodology along with the results and analysis of the resulting dataset. We also provide a summary of the previous work done in past years to create annotated datasets for metaphor identification.

[1] We are not going to address it here as it is beyond the scope of this paper.

2 Related Work

In this section, we will discuss the most relevant research in terms of the dataset preparation and the annotation of linguistic metaphors. As discussed in the previous section, there are several factors that affect the dataset creation and the annotation scheme, including the task definition and the targeted application, which push the dataset creation towards a specific domain or text type. Past research in this area has focused on formal well-structured text such as news or has only targeted a selected examples of metaphors. The majority of researchers formulate their own annotation guidelines and definition of metaphor. One of the main challenges of this work is to introduce an annotation scheme that results in an expert annotated dataset for metaphor identification that have large coverage of metaphoric usages and text genres while maintaining high accuracy. Table 1 provides a detailed summary of the datasets annotated for linguistic metaphors.

TroFi Example Base [1] is one of the earliest metaphor datasets which consists of 3,737 manually annotated English sentences extracted from the 1987-1989 Wall Street Journal corpus (WSJ) covering the literal and metaphoric senses of 50 selected verbs. The dataset has been frequently used to evaluate approaches for metaphor analysis, however there is no information available regarding the inter-annotator agreement (IAA), so its value is questionable. Turney et al. [32] created a dataset of 100 sentences from the Corpus of Contemporary American English (COCA) [5] focusing on metaphoric adjectives. The dataset contains five selected adjectives forming twenty adjective-noun pairs which were manually annotated by five annotators.

Steen [30] employed the metaphor identification procedure (MIPVU) to annotate metaphors in a subset of the British National Corpus (BNC) [2], namely BNC Baby, in order to create the VU Amsterdam Metaphor Corpus (VUA) which has become one of the most popular existing metaphor datasets nowadays. The corpus consists of randomly selected texts from various text genres. Their collaborative annotation scheme annotates metaphors on the word-level, regardless of the word's syntactic type, considering a word as a metaphor as long as its most basic meaning, derived from corpus-based dictionaries, contradicts its current contextual meaning. The basic meaning is typically the most physical or concrete meaning which does not have to be the first sense listed under a word entry. The MIPVU employs two other dictionaries in addition to the corpus-based dictionary. The IAA was measured in terms of Fleiss' kappa [9] among four annotators which averaged 0.84. One of the issues with this procedure is that the sense of every word in the text is considered as a potential metaphor, even idioms or fixed collocations, which are considered inseparable lexical units. Moreover, the annotators have to go through a series of complex decisions starting from chunking the given text into lexical units, then discerning their basic meaning, and finally the metaphoric classification. The uniformity of the basic meaning interpretation may vary from one annotator to the other. Shutova and Teufel [27] adopted the MIPVU annotation scheme, with some modifications, to annotate linguistic metaphors on the word-level focusing on verbs in a subset of the BNC. The corpus comprises 761 sentences and 13,642 words. The IAA was evaluated by means of κ [29] which averaged 0.64 among three native annotators. The authors reported that the conventionality of some metaphors is a source of disagreement. A subset of the VUA corpus comprises around 5,000 verb-object pairs has been prepared in [34]. The adapted VUA subset is drawn from the training verbs dataset from the VUA corpus provided by the NAACL 2018 Metaphor Shared Task[2]. The authors retrieved the

[2] https://github.com/EducationalTestingService/metaphor/tree/master/NAACL-FLP-shared-task

original sentences of around 17,240 annotated verbs, which yielded around 8,000 sentences. Then the verb-direct object relations were extracted using the Stanford parser [4]. The classification of each verb-noun pair was decided based on the metaphoric classification of the verb provided in the original corpus.

Hovy et al. [15] created a dataset by extracting sentences from the Brown corpus [10] to identify metaphors of any syntactic structure on the word-level. They used a list of 329 predefined metaphors as seed to extract sentences that contain the specified expressions. The dataset is manually annotated using crowd-sourcing through Amazon Mechanical Turk (MTurk) platform. The annotators were asked whether a highlighted word in a sentence was used metaphorically or not based on its original meaning. This approach is similar to ours but we annotated metaphoric expressions on the phrase-level focusing on verb-noun pairs. The IAA among seven annotators was 0.57. The annotated instances were filtered out yielding a final corpus consisting of 3,872 instances, out of which 1,749 contains metaphors. Mohler et al. [21] created a dataset focusing on linguistic metaphors in the governance domain. The dataset consists of 500 documents (∼21,000 sentences) manually annotated by three annotators which were extracted from political speeches, websites, and online newspapers. In 2016, the Language Computer Corporation (LCC) annotated metaphor datasets [22] was introduced. The English dataset was extracted from the ClueWeb09 corpus[3]. The freely available part of the dataset contains ∼7,500 metaphoric pairs of any syntactic structure annotated by adopting the MIPVU scheme. There is no clear information regarding the number of annotators or the final IAA of this subset. Tsvetkov et al. [31] created a dataset of ∼2,000 adjective-noun pairs which were selected manually from collections of metaphors on the Web. This dataset is commonly known as the TSV dataset and is divided into 1,768 pairs as a train set and 222 pairs as a test set. An IAA of 0.76 was obtained among five annotators on the test set. The annotators were asked to use their intuition to define the non-literal expressions.

Mohammad et al. [20] annotated different senses of verbs in WordNet [8] for metaphoricity. Verbs were selected if they have more than three senses and less than ten senses yielding a total of 440 verbs. Then the example sentences from WordNet for each verb were extracted and annotated by 10 annotators using crowd-sourcing through the CrowdFlower platform (currently known as Figure Eight). The verbs that were tagged by at least 70% of the annotators as metaphorical or literal were selected to create the final dataset. The dataset consists of 1,639 annotated sentences out of which 410 were metaphorical and 1,229 literal. This dataset, commonly known as the MOH dataset, had been used to model and evaluate systems identifying metaphoric verbs on the word-level. A subset of the MOH dataset has been adapted in [26] to extract the verb-subject and verb-direct object grammar relations, in order to model computational approaches that analyse phrase-level metaphors of verb-noun pairs. The final dataset consists of 647 verb-noun pairs out of which 316 instances are metaphorical and 331 instances are literal.

In an attempt to detect metaphors in social media, Jang et al. [16] acquired a dataset of 1,562,459 posts from an online breast cancer support group. A set of eight predefined words, that can appear either metaphorically or literally in the corpus, were employed to classify each post. An IAA of 0.81 was recorded in terms of Fleiss' kappa among five annotators on MTurk who were provided by a Wikipedia definition of metaphor. Twitter datasets of a figurative nature were prepared in the context of the SemEval 2015 Task 11 on Sentiment Analysis of Figurative Language in Twitter [12]. This dataset is referred to here as the SemEval 2015

[3] https://lemurproject.org/clueweb09/

SAFL dataset. The dataset is originally designed to support the classification and sentiment analysis of tweets containing irony, sarcasm, and metaphors. The available training, and test sets were collected based on lexical patterns that indicate each phenomenon such as using the words "figuratively" and "literally" as lexical markers to collect the metaphoric tweets. Shutova et al. [28] studied the reliability of such technique and discussed that the dependence on lexical markers as a signal of metaphors is not sufficient. The training dataset contains 2,000 tweets which the organisers categorised as metaphoric tweets. We manually annotated a subset of arbitrary selected 200 tweets of the training dataset for use in our preliminary experiments.

Recently, Parde and Nielsen [23] exploited the VUA corpus to create a dataset of phrase-level metaphors annotated for novelty. In this work, 18,000 metaphoric word pairs of different syntactic structures have been extracted from the VUA corpus. Five annotators were then asked to score the highlighted metaphoric expression in a given context for novelty in a scale from 1 to 3. The annotation experiment was set up on MTurk and an IAA of 0.435 was achieved. Another work that addresses metaphor annotation for novelty is [6] focusing on word-level metaphors. Similar to [23], the authors exploited the VUA corpus to annotate 15,180 metaphors for novelty using crowd-sourcing. Different annotation experiments were set up on MTURK to decide: 1) the novelty and conventionality of a highlighted word, 2) the scale of novelty of a given metaphor, 3) scale of "unusualness" of a highlighted word given its context, and 4) the most novel and the most conventionalised metaphor from given samples . The annotators obtained an IAA of 0.39, 0.32 and 0.16 in terms of Krippendorff's alpha for the first three tasks, respectively.

3 Data Preparation

Our aim is to prepare a high-quality annotated dataset focusing on balance, coverage, and representativeness. These factors [7] are central to building a corpus so we considered them besides the other factors discussed earlier. In this section, we discuss the data sources and the preparation steps for creating a dataset annotated for metaphor in tweets.

3.1 Sources

In order to avoid targeting specific topic genres or domains, we utilised two data sources to prepare our dataset which represents two categories of tweets. The first category is general domain tweets which is sampled from tweets pertaining to sentiment and emotions from the SemEval 2018 Task 1 on Affect in Tweets [19]. The second category of data is of a political nature which is sampled from tweets around Brexit [13].

Emotional Tweets. People tend to use figurative and metaphoric language while expressing their emotions. This part of our dataset is prepared using emotion related tweets covering a wide range of topics. The data used is a random sample of the Distant Supervision Corpus (DISC) of the English tweets used in the SemEval 2018 Task 1 on Affect in Tweets, hereafter SemEval 2018 AIT DISC dataset[8]. The original dataset is designed to support a range of emotion and affect analysis tasks and consists of about 100 million tweets[9] collected using emotion-related hashtags such as *"angry, happy, surprised, etc"*. We

[8] available online on: `https://competitions.codalab.org/competitions/17751#learn_the_details-datasets`

[9] Only the *tweet-id*s were released and not the tweet text due to copyright and privacy issues.

Table 1 Summary of the datasets created for linguistic metaphor identification. *The dataset is not directly available online but can be obtained by contacting the authors.

	Level of Analysis	**Syntactic Structure**	**text type**	**domain**	**crowd-source**	**IAA**	**annot-ators**	**size**	**available**
Birke and Sarkar, 2006 (TroFi Example Base)	word-level	verb	selected examples (News)	open	no	-	-	3,737 sentences	yes[4]
Steen, 2010 (VUA)	word-level	any	known-corpus (The BNC)	open	in-house	0.84	4	∼200,000 word (∼16,000 sentences)	yes[5]
Shutova et al., 2010	word-level	verb	known-corpus (The BNC)	open	in-house	0.64	3	761 sentences	yes*
Turney et al., 2011	word-level	verb adjective	selected examples (News)	open	no	-	5	100 sentences	no
Hovy et al., 2013	word-level	any	known-corpus (The Brown Corpus)	open	yes*	0.57	7	3,872 instances	no
Mohler et al., 2013	word-level	any	selected examples	restricted (governance)	no	-	-	21,000 sentences	no
Tsvetkov et al., 2014 (TSV)	phrase-level	adj–noun	selected examples (News)	open	no	-	-	2,000 adj-noun pairs	yes[6]
Jang et al. 2015	word-level	noun	selected examples (Social Media)	restricted (breast cancer)	yes	0.81	5	2,335 instances	no
Mohler et al., 2016 (LCC)	word-level	any	known-corpus (ClueWeb09)	open	no	-	-	7,500 metaphoric pairs	partially
Mohammad, 2016 (MOH)	word-level	verb	selected examples (WordNet)	open	yes	-	10	1,639 sentences	yes[7]
Shutova, 2016 (adaptation of MOH)	phrase-level	verb-noun	selected examples (WordNet)	open	-	-	-	-	yes*
Our dataset	phrase-level	verb-direct obj	tweets	open	yes	0.70-0.80	5	$\tilde{2}$500 tweets	yes*

[4] http://natlang.cs.sfu.ca/software/trofi.html
[5] http://ota.ahds.ac.uk/headers/2541.xml
[6] https://github.com/ytsvetko/metaphor
[7] http://saifmohammad.com/WebPages/metaphor.html

retrieved the text of around 20,000 tweets given their published *tweet-ids* using the Twitter API[10]. We preprocessed the tweets to remove URLs, elongations (letter repetition, e.g. verrrry), and repeated punctuation as well as duplicated tweets then arbitrary selected around 10,000 tweets.

Political Tweets. Metaphor plays an important role in political discourse which motivated us to devote part of our dataset to political tweets. Our goal is to manually annotate tweets related to the Brexit referendum for metaphor. In order to prepare this subset of our dataset, we looked at the Brexit Stance Annotated Tweets Corpus[11] introduced by Grčar et al. [13]. The original dataset comprises 4.5 million tweets collected in the period from May 12, 2016 to June 24, 2016 about Brexit and manually annotated for stance. The text of around 400,000 tweets on the referendum day is retrieved from the published *tweet-ids*. These tweets contained a lot of duplicated tweets and re-tweets. We cleaned and preprocessed them similar to the emotional tweets as discussed above yielding around 170,000 tweets.

3.2 Initial Annotation Scheme

We suggested a preliminary annotation scheme and tested it through an in-house pilot annotation experiment before employing crowd-sourcing. In this initial scheme, the annotators are asked to highlight the words which are used metaphorically relying on their own intuition, and then mark the tweet depending on metaphor presence as *"Metaphor"* or *"NotMetaphor"*. In this experiment, 200 tweets were extracted from the SemEval 2015 SAFL dataset mentioned in Section 2. The tweets are sarcastic and ironic in nature due to the way they were initially collected by querying Twitter Search API for hashtags such as *"#sarcasm, #irony"*. The annotation is done by three native speakers of English from Australia, England, and Ireland. The annotators were given several examples to explain the annotation process. We developed a set of guidelines for this annotation experiment in which the annotators were instructed to, first, read the whole tweet to establish a general understanding of the meaning. Then, mark it as metaphoric or not if they suspect that it contains a metaphoric expression(s) based on their intuition taking into account the given definition of a metaphor. A tweet might contain one or more metaphors or might not contain any metaphors. Finally, the annotators were asked to highlight the word(s) that according to their intuition has a metaphorical sense.

The annotators achieved an inter-annotator agreement of 0.41 in terms of Fleiss' kappa. Although the level of agreement was quite low, this was expected as the metaphor definition depends on the native speaker's intuition. The number of annotated metaphors varies between individual annotators with maximum metaphors' percentage of 22%. According to the annotators, the task seemed quite difficult and it was very hard to pick the boundary between metaphoric and literal expressions. A reason for this is perhaps the ironic nature of the tweets, with some authors deliberately being ambiguous. Sometimes the lack of background knowledge adds extra complexity to the task. Another important challenge is the use of highly conventionalised language. The question that poses itself here is how to draw a strict line about which expression should be considered as a metaphor and which is not.

We concluded from this initial experiment that the annotation task is not ready for crowd-sourcing due to the previously mentioned limitations. It would be still an expensive task in terms of the time and effort consumed by the annotators. We explored the usage of

[10] Twitter API: `https://developer.twitter.com/en/docs/api-reference-index`
[11] available online on: `https://www.clarin.si/repository/xmlui/handle/11356/1135`

WordNet as a reference for sense distinction on 100 tweets. An IAA agreement of 0.21 was achieved which is extremely low due to the annotators' disagreement on what they believed to be metaphors in their initial judgement, therefore they checked WordNet for different expressions. This initial pilot study also revealed that this dataset is not suitable for the annotation so we changed it as will be discussed in sub-section 3.1 to help improve the accuracy of the annotations.

3.3 Weakly Annotated dataset

In order to address the limitations of the initial annotation experiments, we prepared a weakly annotated dataset using a metaphor identification system, to reduce the cognitive load for annotators and maintain consistency. This system will be used to identify potential metaphoric expressions in tweets. Then, crowd-sourcing will be employed to ask a number of annotators to identify the expressions that are metaphorical in their judgement from these pre-identified ones. This way, the cognitive load on the annotators will be reduced while maintaining consistency. Figure 1 shows the process of creating our dataset.

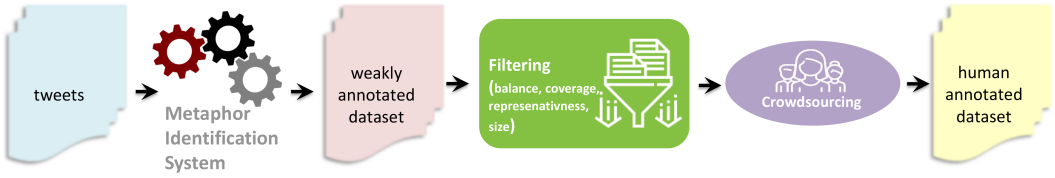


Figure 1 The proposed approach to create a dataset of tweets for metaphor identification.

Zayed et al. [34] introduced a weakly supervised system which makes use of distributed representations of word meaning to capture metaphoricity focusing on identifying verb-noun pairs where the verb is used metaphorically. The system extracts verb-noun pairs using the Stanford parser [4]. Then pre-trained word embeddings models are employed to measure the semantic similarity between the candidate pair and a predefined seed set of metaphors. The given candidate is classified using a previously optimised similarity threshold. We used this system to prepare a weakly annotated dataset using the data discussed in sub-section 3.1. The reason behind choosing this system is that it employs fewer lexical resources and does not require annotated datasets. Moreover, it is a weakly supervised system that employs a small seed set of predefined metaphors and is not trained on a specific dataset or text genre.

The arbitrarily selected tweets from both the emotional tweets and the political tweets subsets are used individually as input to the system which highlights the verb-direct object pairs using a parser as potential candidates for metaphor classification. A candidate is classified as a metaphor or not by measuring its semantic similarity to a predefined small seed set of metaphors which acts as an existing known metaphors sample. Metaphor classification is performed based on a previously calculated similarity threshold value. The system labelled around 42% and 48% as metaphorical expressions of the candidates from the emotional tweets subset and the political tweets subset respectively.

3.4 Dataset Compilation

Now that we have two weakly annotated subsets of emotional and political tweets, our approach for selecting the final subset of each category of tweets is driven by achieving the following factors:

1. **Balance**: the dataset should equally represent positive and negative examples.
2. **Verbs Representativeness (Verb Coverage)**: the dataset should cover a wide range of verbs and a variety of associated nouns.
3. **Sense Coverage**: ideally each verb should appear at least once in its metaphoric sense and once literally. If the verb does not have one of these senses, more examples should be included. Moreover, unique object arguments of each verb should be represented.
4. **Size**: to ensure usability in a machine learning setting, the dataset should be sizeable.

To ensure verbs representativeness, we employed a set of 5,647 verb-object pairs from the adapted subsets of the MOH dataset (647 verb-direct object pairs) [26] and the VUA corpus (exactly 4,526 verb-direct object pairs) [34]. For each verb in the set[12], all the tweets that contain this verb are extracted without regard to the associated noun (object) argument or the initial metaphoric/literal classification of the weakly supervised system. This step yielded around 3,000 instances from the emotional tweets subset and 38,000 instances from the political tweets subset. For each verb, we randomly selected at least one metaphoric instance and one literal instance using the initial classification by the system to ensure balance, e.g. *"find love"* vs *"find car key"* and *"send help"* vs *"send email"*. Also we ensured the uniqueness of the noun argument associated with each target verb to ensure sense coverage within each subset and across both subsets meaning that the same verb appearing in both subsets has different nouns in order to cover a lot of arguments. Each instance should not exceed five words such as *"send some stupid memory"* or *"abandon a humanitarian approach"*. We observed that the parser more frequently made errors on these longer phrases and thus removing them eliminated many erroneous sentences. Moreover, from preliminary experiments, we realised that the annotators got confused when there are multiple adjectives between the verb and the direct object in a given expression and focused on them instead of the object. Although it might be argued that we could have selected a random set of the tweets but this will not achieve our goal of verb and sense coverage. Moreover, another approach to ensure verb representativeness would have been employing VerbNet [24] but we wanted to be sure that the majority of selected verbs have metaphoric usages.

Our final dataset comprises around 2,500 tweets of which around 1,100 tweets are emotional tweets of general topics and around 1,400 tweets are political tweets related to Brexit. Each tweet has a highlighted verb-object expression that need to be classified according to the metaphoricity of the verb given the accompanying noun (direct object) and the given context. Our next step is to employ crowd-sourcing to manually annotate these expressions. Table 2 shows examples of the different instances appeared in the emotional and political tweets subsets.

4 Annotation Process

4.1 Metaphor Definition

In this work, we adopt the most well-known definition of metaphor which is the conceptual metaphor theory (CMT) introduced by Lakoff and Johnson [17]. Therefore, we view a word

[12] The number of unique verbs (lemma) in this set is 1,134 covering various classes.

Table 2 Examples of the instances appearing in the emotional and political tweets subsets and the corresponding classification of the employed weakly supervised system. *The human annotation disagrees with the system annotation on these examples.

Emotional Tweets	System Classification	Political Tweets	System Classification
accept the fact	metaphor	add financial chaos	not metaphor*
attract hate	metaphor	back #brexit cause	metaphor
break ego	not metaphor*	blame heavy rain	not metaphor
deserves a chance	metaphor*	claim back democracy	metaphor
have time	metaphor	claiming expenses	metaphor*
bring happiness	metaphor	have a say	metaphor
hold phone	not metaphor	hand over britain	not metaphor*
join team	not metaphor	make history	metaphor
win game	not metaphor	write your vote	not metaphor

or an expression as metaphoric if it has at least one basic/literal sense and a secondary metaphoric sense. The literal sense is more concrete and used to perceive a familiar experience while the metaphoric sense is abstract. Moreover, we consider Hank's [14] view that the metaphoric sense should resonate semantically with the basic sense which means that the metaphorical sense corresponds closely with the literal sense sharing similar semantic features. For example, the metaphoric expression *"launch a campaign"* aligns with (resonates with) more literal, more concrete expressions such as *"launching a boat"*. In this work, we are interested in analysing verb-noun pairs where the verb could be used metaphorically and the noun is a direct object. Research has shown that the majority of metaphoric expressions clusters around verbs and adjectives [25]. We made some distinctions as follows:

Idioms and Similes. We make a distinction between metaphors and other figures of speech that they might be confused with, namely idioms and similes. Idioms such as *"blow the whistle, call the shots, pull the rug out, turn a blind eye, etc."* were filtered manually.

Verbs with No Metaphorical Potential. We excluded auxiliary and model verbs from our dataset assuming that they exhibit no potential of being used metaphorically.

Verbs with Weak Metaphorical Potential. In addition to verbs that exhibit strong potential of being metaphors, we are interested in investigating the metaphoricity of light verbs such as *"do, get, give, have, make, take"* and aspectual verbs such as *"begin, end, finish, start, stop"* as well as other verbs such as *"accept, choose, cause, remember, etc"*. Section 5 presents an analysis of these verbs as they appeared in the proposed dataset. In order to ensure balance, our dataset contains verbs that exhibit both strong and weak metaphorical potential.

4.2 Annotation Task

The annotation task is concerned with the identification of linguistic metaphors in tweets. The main goal is to discern the metaphoricity of a target verb in a highlighted verb-object expression in a given tweet. We set up our annotation task on Amazon Mechanical Turk (MTurk). Five native English speakers were hired to annotate the dataset whose field of study is bachelor of arts with either English, journalism or creative writing.

Task Definition. The annotators were asked to review the tweets and classify the highlighted expression as being used metaphorically or not, based on the provided definition of metaphor and their intuition of the basic sense of the verb.

Guidelines. Each tweet has a highlighted expression of a verb-object (noun) expression. The annotators were instructed to follow a set of guidelines in order to classify the highlighted expression, which are:

1. Read the whole tweet to establish a general understanding of the meaning.
2. Determine the basic meaning of the verb in the highlighted expression. Then, examine the noun (object) accompanying the verb and check whether the basic sense of the verb can be applied to it or not. If it can not, then the verb is probably used metaphorically.
3. Select how certain they are about their answer.

These steps were represented in the task as three questions appearing to the annotators on MTurk as shown in Figure 2.

Reading the whole tweet is important as giving a decision based on reading the highlighted expression only is not enough and leads to inaccurate results. The annotators can skip the tweet if they do not understand it but we set a threshold for skipping tweets. If the annotator is confused about whether an expression is a metaphor or not they were asked to select the "don't have a clue" option in question 3. The annotators were encouraged to add some notes regarding their confusion or any insights they would like to share. We provided the annotators with several examples to explain the annotation process and to demonstrate the definition of metaphor adopted by this work as well as showing how to discern the basic sense of a verb.

Task Design. We created three annotation tasks on MTurk. The first task is a demo task of 120 tweets from the emotional tweets subset. These tweets included 20 gold tweets with known answers which were obtained by searching the emotional tweets subset for metaphoric expressions (positive examples) from the MOH dataset as well as including some negative examples. This task acted as a training demo to familiarise the annotators with the platform and to measure the understanding of the task. Moreover, it acted as a test for selecting the best preforming annotators among all applicants. The efficiency of each applicant is measured in terms of: 1) the time taken to finish the task, 2) the amount of skipped questions and 3) the quality of answers which is measured based on the gold tweets. We selected the top five candidates to proceed with the main tasks. The second task is the annotation of the emotional tweets subset and the third task was devoted to annotating the political tweets subset.

We designed our tasks as pages of 10 tweets each. Pages are referred to as a human intelligence tasks (HITs) by MTurk and annotators were paid per HIT (page). We

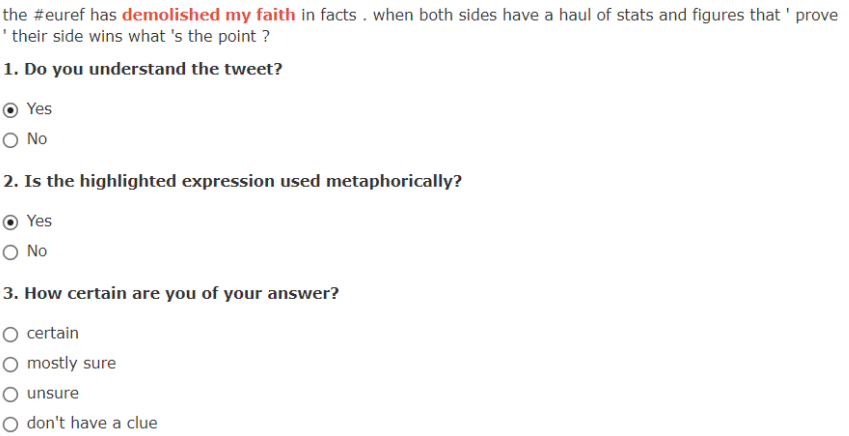


Figure 2 A screenshot of the questions in the annotation task given to the annotators on MTurk.

estimated the time taken to annotate around 200 tweets to be one hour; therefore, we paid 60 cents for each page. This comes down to $12 per hour, which aligns with the minimum wage regulations of the country where the authors resided at the time of this publication.

4.3 Evaluation

Inter-annotator Agreement. The inter-annotator agreement (IAA) evaluation was carried out in terms of Fleiss' kappa between the five annotators as shown in Table 3. As discussed earlier, we wanted to have a deeper look into light and aspectual verbs, as well as verbs with weak metaphoric potential, so we computed the IAA with and without these verbs for each subset of our dataset. As observed from the results, the annotators were able to achieve a substantial agreement (as for Landis an Koch [18] scale) on the demo task as well as the emotional tweets and the political tweets subsets. After the demo task, the annotators were instructed to pay extra attention to light verbs and to be consistent with similar abstract nouns as much as they can, meaning that "give hope" would often have the same annotation as *"give anxiety/faith"*. To ensure better performance and avoid distraction, we advised the annotators to annotate around 300 tweets per day and resume after reading the instructions again. Since we did not control this rule automatically, we verified that all annotators adhered to this rule by manually checking the time stamps of the annotated tweets.

Table 3 Inter-Annotator Agreement between the five annotators using Fleiss' kappa. The excluded verbs are light verbs, aspectual verbs, in addition to weak metaphoric potential verbs including "accept, choose, enjoy, imagine, know, love, need, remember, require, want".

	Fleiss' kappa		
	partial exclusion (keep light verbs)	**full exclusion**	**no exclusion**
Demo Task (120 tweets)	0.627 (106 annotated instances)	0.715 (85 annotated instances)	0.623 (108 annotated instances)
Emotional Tweets (1,070 tweets)	0.742 (884 annotated instances)	0.732 (738 annotated instances)	0.701 (1,054 annotated instances)
Political Tweets (1,391 tweets)	0.806 (1,341 annotated instances)	0.805 (1,328 annotated instances)	0.802 (1,389 annotated instances)

Points of (Dis-)agreement. Tables 4 and 5 lists examples of the agreements and disagreements between the five annotators. The majority of disagreements centred around light verbs and verbs with weak metaphoric potential. The next section discusses the annotation results in detail and presents the statistics of the dataset.

5 Dataset Statistics and Linguistic Analysis

5.1 Statistics

The statistics of each subset of the dataset is presented in Table 6 focusing on different statistical aspects of our dataset. It is worth mentioning that the political tweets subset contains 431 more unique verbs that did not appear in the emotional tweets subset. The text of the political tweets was more understandable and structured. The emotional tweets subset contains some tweets about movies and games that sometimes the annotators found hard to understand.

Table 4 Examples of agreements among the five annotators (100% majority vote).

		majority vote
Emotional Tweets	its great to be happy, but its even better to **bring happiness** to others.	metaphor
	make memories you will look back and smile at.	
	as long as the left stays so ugly, bitter, annoying & unlikeable, they will not **win any elections**...	not metaphor
	they forget this when they **have money** and start tweeting like they have all the answers	
Political Tweets	make or **break moment** today! together we are stronger! vote remain #strongerin #euref	metaphor
	...cameron can not win this #euref without your support. how many will **lend their support** to...	
	person's details taken by police for offering to **lend a pen** to voters, what a joke.	not metaphor
	in just a couple of days, no one will ever have to **utter the word** 'brexit' ever again	

Table 5 Examples of disagreements among the five annotators (60% majority vote).

		majority vote
Emotional Tweets	someone should **make a brand** based off of triangle noodles that glow in the dark. call it illuminoodle...	metaphor
	smile for the camera, billy o. if you **need a smile** every day then #adoptadonkey @donkeysanctuary	
	cities are full of mundane spaces. **imagine the potential** to transform them into catalysts for positive emotions	not metaphor
	our captors are treating us well and we are very happy and well **enjoying their kind hospitality**	
Political Tweets	perhaps we can **achieve a cohesive society** when the referendum is over, but it does not feel like that is possible. #euref	metaphor
	#euref conspiracy theories **predict people's voting intentions**. will they sway today's vote?	
	democracy works there's still time. british people can not be bullied do not **believe the fear** #voteleave	not metaphor
	what's interesting here is not the figure but that it was from an online poll which has always **favoured the leave**.	

Table 6 Statistics of the proposed dataset of tweets.

	Demo Task	Emotional Tweets	Political Tweets
# of tweets	120	1,070	1,390
# of unique verb-direct object (noun) pairs	119	1,069	1,390
Average tweet length	23.82	22.14	21.12
# of unique verbs (lemma) (in the annotated verb-noun pairs)	71	321	676
# of unique nouns (in the annotated verb-noun pairs)	102	725	706
% instances annotated as metaphors	63.15%	50.47%	58.16%
% instances annotated as not metaphors	36.84%	49.54%	41.84%
% instances annotated with agreement majority vote of 60%	20.17%	10.39%	12.29%
# of non-understandable tweets (skipped)	5.2%	1.68%	0.14%

5.2 Linguistic Analysis

As observed from the IAA values listed in Table 3, light and aspectual verbs as well as some other verbs represent a major source of confusion among the annotators. Although other researchers pointed out that they exhibit low potential of being metaphors and excluded them from their dataset, our dataset covers different examples of these verbs with different senses/nouns. The majority vote of the annotators on such cases could give us some insight on the cases where these verbs can exhibit metaphorical sense.

In the following paragraphs, we provide a linguistic analysis of the proposed dataset performed by manual inspection. The majority of annotators tend to agree that the verb *"have"* exhibits a metaphoric sense when it comes with abstract nouns such as *"anxiety, hope, support"* as well as other arguments including *"meeting, question, theory, time, skill, vote"*.

On the other hand, it is used literally with nouns such as *"clothes, friend, illness, license, money"*. The annotators find the light verbs *"get, give, and take"* to be more straightforward while discerning their metaphoric and literal usages. They agreed on their metaphorical usage with abstract nouns such as *"chance, happiness, smile, time, victory"* and their literal usage with tangible concepts including *"food, job, medication, money, notification, results"*. Regarding the verb *"make"* the annotators agreed that as long as the accompanied noun cannot be *crafted* then it is used metaphorically. Metaphors with this verb include *"difference, friends, money, progress, time"*, while literal ones include *"breakfast, mistake, movie, noise, plan"*.

The nouns occurring with the verb *"start"* in metaphoric expressions include *"bank, brand, friendship"*. Moreover, there are some rhetorical expressions such as *"start your leadership journey/living/new begining"*. The nouns appearing in the expressions classified as literal include *"argument, car, course, conversation, petition"*. The verb "end" occurred with "horror, feud" metaphorically and "thread, contract" literally according to the majority vote.

The annotators agreed that nouns such as *"food, hospitality, life, music"* occurring with the verb *"enjoy"* form literal expressions while the only metaphoric instance is *"enjoy immunity"*. In the case of the verb *"love"*, the majority of annotators agreed that it is not used metaphorically as you can love/hate anything with no metaphorical mapping between concepts. The disagreements revolve around the cases when the expression is an exaggeration or a hyperbole e.g. *"love this idea/fact/book"*. Expressions have stative verbs of thought such as *"remember and imagine"* are classified as non-metaphoric. The only debate was about the expression *"...remember that time when..."* as, according to the annotators, it is a well-known phrase (fixed expression). We looked at the verbs *"find and lose"* and they were easy to annotate following the mapping between abstract and concrete senses. They are classified as metaphors with abstract nouns such as *"love, opportunity, support"* as well as something virtual such as *"lose a seat (in the parliament)"*. However, it was not the case for the verb *"win"*. The majority agreed that expressions such as *"win award/election/game"* are literal expressions while the only disagreement was on the expression *"win a battle"* (3 annotators agreed that it is used metaphorically).

Annotating the verbs *"accept, reject"* was intriguing as well. The majority of annotators classified the instances *"accept the fact/prices"* as literal while in their view *"accept your past"* is a metaphor. An issue is raised regarding annotating expressions that contain the verbs *"cause, blame, need, want"*. Most agreed that *"need apology/job/life"* can be considered as metaphor while *"need date/service"* is not.

From this analysis, we conclude that following the adopted definition of metaphor helped the annotators to discern the sense of these verbs. Relying on the annotators' intuition (guided by the given instructions) to decide the basic meaning of the verb led to some disagreements but it was more time and effort efficient than other options. Light verbs are often used metaphorically with abstract nouns. There are some verbs exhibiting weak metaphoric potential and classifying them is not as straightforward as other verbs. However, they might be used metaphorically on occasions, but larger data is required to discern these cases and find a solid pattern to define their metaphoricity. Hyperbole and exaggerations and other statements that is not meant to be taken literally need further analysis to discern its metaphoricity. Sharing and discussing the disagreements after each annotation task among the annotators helped to have a better understanding of the task.

6 Conclusion

This work proposes an annotation methodology to create a high-quality dataset of tweets annotated for metaphor using crowd-sourcing. Our approach is driven by achieving balance, sense coverage and verbs representativness as well as high accuracy. We were able to introduce a better quality annotation of metaphors in spite of the conventionality of metaphors in our everyday language compounded by the challenging context of tweets. The employed approach resulted in a dataset of around 2,500 tweets annotated for metaphor achieving a substantial inter-annotator agreement despite the difficulty of defining metaphor. Although, we focused on annotating verb-direct object pairs of linguistic metaphors in tweets, this approach can be applied to any text type or level of metaphor analysis. The annotation methodology relies on an existing metaphor identification system to facilitate the recognition and selection of the annotated instances by initially creating a weakly annotated dataset. This system could be substituted by any other model to suit the type of targeted metaphors in order to reduce the cognitive load on the annotators and maintain consistency. Our dataset consists of various topic genres focusing on tweets of general topics and political tweets related to Brexit. The dataset will be publicly available to facilitate research on metaphor processing in tweets.

We are planning to use this dataset to create a larger dataset of tweets annotated for metaphors using semi-supervised methods. Additionally, an in-depth qualitative and quantitative analysis will be carried out to follow up on the conclusions that have been drawn in this work. Furthermore, we are interested in having a closer look at the metaphors related to Brexit on Twitter. We are also interested in investigating the metaphoric sense of verbal multi-word expressions (MWEs) by looking into the dataset released as part of the PARSEME shared-task [33].

References

1. Julia Birke and Anoop Sarkar. A clustering approach for nearly unsupervised recognition of nonliteral language. In *In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '06, pages 329–336, Trento, Italy, April 2006.
2. Lou Burnard. About the British National Corpus, 2009. URL: `http://www.natcorp.ox.ac.uk/corpus/index.xml`.
3. Jonathan Charteris-Black. Metaphor in Political Discourse. In *Politicians and Rhetoric: The Persuasive Power of Metaphor*, pages 28–51. Palgrave Macmillan UK, London, 2011.
4. Danqi Chen and Christopher Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 740–750, Doha, Qatar, October 2014.
5. Mark Davies. The 385+ million word Corpus of Contemporary American English (1990–2008+): Design, architecture, and linguistic insights. *International Journal of Corpus Linguistics*, 14(2):159–190, 2009.
6. Erik-Lân Do Dinh, Hannah Wieland, and Iryna Gurevych. Weeding out Conventionalized Metaphors: A Corpus of Novel Metaphor Annotations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 1412–1424, Brussels, Belgium, November 2018.
7. David Evans. Compiling a corpus. *Corpus building and investigation for the Humanities*, 2007 (accessed December 23, 2018). URL: `https://www.birmingham.ac.uk/Documents/college-artslaw/corpus/Intro/Unit2.pdf`.
8. Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
9. Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

10 W. Nelson Francis and Henry Kucera. The Brown Corpus: A Standard Corpus of Present-Day Edited American English. Technical report, Brown University Liguistics Department, 1979.

11 Dedre Gentner, Brian Bowdle, Phillip Wolff, and Consuelo Boronat. Metaphor Is Like Analogy. In D. Gentner, K. J. Holyoak, and B. N. Kokinov, editors, *The analogical mind: Perspectives from cognitive science*, pages 199–253. The MIT Press, Cambridge, MA, USA, 2001.

12 Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 470–478, Denver, CO, USA, June 2015.

13 Miha Grčar, Darko Cherepnalkoski, Igor Mozetič, and Petra Kralj Novak. Stance and influence of Twitter users regarding the Brexit referendum. *Computational Social Networks*, 4(6):1–25, July 2017.

14 Patrick Hanks. Three Kinds of Semantic Resonance. In *Proceedings of the 17th EURALEX International Congress*, pages 37–48, Tbilisi, Georgia, September 2016.

15 Dirk Hovy, Shashank Srivastava, Sujay Kumar Jauhar, Mrinmaya Sachan, Kartik Goyal, Huiying Li, Whitney Sanders, and Eduard Hovy. Identifying Metaphorical Word Use with Tree Kernels. In *Proceedings of the 1st Workshop on Metaphor in NLP*, pages 52–56, Atlanta, GA, USA, June 2013.

16 Hyeju Jang, Seungwhan Moon, Yohan Jo, and Carolyn Rose. Metaphor detection in discourse. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '15, pages 384–392, Prague, Czech Republic, September 2015.

17 George Lakoff and Mark Johnson. *Metaphors we live by.* University of Chicago Press, Chicago, USA, 1980.

18 J. Richard Landis and Gary G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174, 1977.

19 Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. SemEval-2018 Task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, SemEval '18, pages 1–17, New Orleans, LA, USA, June 2018.

20 Saif M. Mohammad, Ekaterina Shutova, and Peter D. Turney. Metaphor as a Medium for Emotion: An Empirical Study. In *Proceedings of the 5th Joint Conference on Lexical and Computational Semantics*, *Sem '16, pages 23–33, Berlin, Germany, 2016.

21 Michael Mohler, David Bracewell, Marc Tomlinson, and David Hinote. Semantic Signatures for Example-Based Linguistic Metaphor Detection. In *Proceedings of the 1st Workshop on Metaphor in NLP*, pages 27–35, Atlanta, GA, USA, June 2013.

22 Michael Mohler, Mary Brunson, Bryan Rink, and Marc Tomlinson. Introducing the LCC Metaphor Datasets. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, LREC '16, pages 4221–4227, Portorož, Slovenia, May 2016.

23 Natalie Parde and Rodney Nielsen. A Corpus of Metaphor Novelty Scores for Syntactically-Related Word Pairs. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, LREC '18, pages 1535–1540, Miyazaki, Japan, May 2018.

24 Karin Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon.* PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 2006.

25 Ekaterina Shutova. Design and Evaluation of Metaphor Processing Systems. *Computational Linguistics*, 41(4):579–623, December 2015.

26 Ekaterina Shutova, Douwe Kiela, and Jean Maillard. Black Holes and White Rabbits: Metaphor Identification with Visual Features. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '16, pages 160–170, San Diego, CA, USA, June 2016.

27 Ekaterina Shutova and Simone Teufel. Metaphor Corpus Annotated for Source-Target Domain Mappings. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, LREC '10, pages 255–261, Malta, May 2010.

28 Ekaterina Shutova, Simone Teufel, and Anna Korhonen. Statistical Metaphor Processing. *Computational Linguistics*, 39(2):301–353, June 2013.

29 S. Siegel and N. Castellan. *Nonparametric statistics for the behavioral sciences*. Mc Graw-Hill, 1988.

30 Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. *A Method for Linguistic Metaphor Identification: From MIP to MIPVU*. Converging evidence in language and communication research. John Benjamins Publishing Company, 2010.

31 Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. Metaphor Detection with Cross-Lingual Model Transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL '14, pages 248–258, Baltimore, MD, USA, June 2014.

32 Peter D. Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 680–690, Edinburgh, Scotland, UK, July 2011.

33 Abigail Walsh, Claire Bonial, Kristina Geeraert, John P. McCrae, Nathan Schneider, and Clarissa Somers. Constructing an Annotated Corpus of Verbal MWEs for English. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions*, LAW-MWE-CxG-2018, pages 193–200, Santa Fe, NM, USA, August 2018.

34 Omnia Zayed, John Philip McCrae, and Paul Buitelaar. Phrase-Level Metaphor Identification using Distributed Representations of Word Meaning. In *Proceedings of the Workshop on Figurative Language Processing*, pages 81–90, New Orleans, LA, USA, June 2018.

Inflection-Tolerant Ontology-Based Named Entity Recognition for Real-Time Applications

Christian Jilek
German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
Department of Computer Science, TU Kaiserslautern, Germany
christian.jilek@dfki.de

Markus Schröder
German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
Department of Computer Science, TU Kaiserslautern, Germany
markus.schroeder@dfki.de

Rudolf Novik
German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
rudolf.novik@dfki.de

Sven Schwarz
German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
sven.schwarz@dfki.de

Heiko Maus
German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
heiko.maus@dfki.de

Andreas Dengel
German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
Department of Computer Science, TU Kaiserslautern, Germany
andreas.dengel@dfki.de

Abstract

A growing number of applications users daily interact with have to operate in (near) real-time: chatbots, digital companions, knowledge work support systems – just to name a few. To perform the services desired by the user, these systems have to analyze user activity logs or explicit user input extremely fast. In particular, text content (e.g. in form of text snippets) needs to be processed in an information extraction task. Regarding the aforementioned temporal requirements, this has to be accomplished in just a few milliseconds, which limits the number of methods that can be applied. Practically, only very fast methods remain, which on the other hand deliver worse results than slower but more sophisticated Natural Language Processing (NLP) pipelines.

In this paper, we investigate and propose methods for real-time capable Named Entity Recognition (NER). As a first improvement step, we address word variations induced by inflection, for example present in the German language. Our approach is ontology-based and makes use of several language information sources like Wiktionary. We evaluated it using the German Wikipedia (about 9.4B characters), for which the whole NER process took considerably less than an hour. Since precision and recall are higher than with comparably fast methods, we conclude that the quality gap between high speed methods and sophisticated NLP pipelines can be narrowed a bit more without losing real-time capable runtime performance.

2012 ACM Subject Classification Computing methodologies → Information extraction; Computing methodologies → Semantic networks

Keywords and phrases Ontology-based information extraction, Named entity recognition, Inflectional languages, Real-time systems

Digital Object Identifier 10.4230/OASIcs.LDK.2019.11

© Christian Jilek, Markus Schröder, Rudolf Novik, Sven Schwarz, Heiko Maus, and Andreas Dengel;
licensed under Creative Commons License CC-BY

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 11; pp. 11:1–11:14

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – DE 420/19-1.

Acknowledgements We thank Sven Hertling, Jörn Hees, Erfan Shamabadi, Oleksii Kotvytskyi and Tim Sprengart for their contributions in this project's early and late phase, respectively.

1 Introduction

The number of application areas, in which users are supported by systems that operate in (near) real-time, grows: chatbots, digital companions, knowledge work support systems – just to name a few. Our targeted scenario involves a system based on Semantic Desktop [15] technology, that semi-automatically re-organizes itself based on user context [10] in order to better support knowledge work and information management activities[1]. We envision an intelligent, proactive assistance parallel to the actual work. Such systems need mechanisms to analyze observed user activities (entering text, browsing a website, reading/writing files, ...) in order to decide on the right support measures and perform them accordingly. The demand for very short reaction times limits the number of methods that can be applied.

In this paper, we focus on Information Extraction (IE) methods, more precisely Named Entity Recognition (NER), that are ontology-based (our system operates on knowledge graphs in the background) and meet the demand for providing meaningful results within only a few milliseconds on users' typical computing devices. By *only a few* we actually mean a small two-digit number of milliseconds. According to Miller (1968) and Card et al. (1991), as cited in [13], 100 ms is "about the limit for having the user feel that the system is reacting instantaneously" and 1000 ms is "about the limit for the user's flow of thought to stay uninterrupted". Our goal is to stay below the first value. In cases, in which this is not possible (e.g. too much data to be processed at once), 1000 ms should be the upper bound of processing time to be tolerated. Since we also need some time for selecting and performing the support measures, the IE task has to be completed within only a fraction of this time span. Such strict temporal requirements usually rule out very sophisticated Natural Language Processing (NLP) pipelines (higher quality solutions but slow), leaving only rather simple (lower quality) but very fast methods often based on pre-defined rules or gazetteers. A gazetteer is conceptually just a list of terms (typically static), that the input text is later scanned for, e.g. the names of persons, organizations or locations. Since our scenario also involves highly inflectional languages like German[2], we additionally have to take slight variations of such terms into account. To vividly illustrate the problem of inflections in NER, we fed the first paragraph of the German Wikipedia article of *Propositional calculus* (German: *Aussagenlogik*) to *DBpedia Spotlight*[3] [11], a well-known and often used recognizer for Wikipedia/DBpedia[4] entities in given text snippets. The results are depicted in Figure 1 (middle section): Twelve entities (in just three sentences; we highlighted them in yellow) are not found, ten of them due to lexical variations induced by inflection. E.g. *Wahrheitswert* (*truth value*) is found, whereas its inflected forms ending with *-e* and *-en* are not. If we lower the confidence to 0.0, there are still some entities missing and false positives come up.

In summary, our goal is to find or implement methods that are fast enough to meet the aforementioned temporal constraints while at the same time achieving better results than standard high speed methods. Recognizing entities despite the just mentioned lexical

[1] for an overview and more details please see `https://comem.ai/`
[2] other inflectional languages: Spanish, Latin, Hebrew, Hindi, Slavic languages, ...
[3] `https://www.dbpedia-spotlight.org/demo/`
[4] `https://wiki.dbpedia.org/`

Figure 1 First paragraph of the German Wikipedia article of *Aussagenlogik* (top) fed to DBpedia Spotlight [11] using confidence values of 0.5 (middle) and 0.0 (bottom). (highlighting we applied: green: existing Wikipedia articles not linked in the original document, yellow: false negatives, red: false positives.)

variations induced by inflection would be a first improvement step. Note that disambiguation as well as recognizing Named Entities (NE) yet unknown to the system (i.e. not available as instances in the knowledge graph) are out of this paper's scope. Since there is a lot of explicitated contextual information available in our system, we intend to address disambiguation in our scenario in a future paper.

The rest of this paper is structured as follows: Section 2 provides an overview of related work in this area. Our approach is described in Section 3 and its evaluation is presented in 4. In Section 5, we conclude this paper and give a an outlook on possible future work.

2 Related Work

We were looking for approaches (more or less) explicitly addressing inflection tolerance or real-time capability, preferably both at the same time:

Concerning real-time capability, Dlugolinsky et al. [8] present an overview of different gazetteer-based approaches, especially referring to various versions included in the GATE (General Architecture for Text Engineering) framework [6]. They distinguish between

character- and token-based variants and state that the latter usually have "longer running time and low processing performance". They thus focus on character-based gazetteers and present several versions [8, 12]. Since some of their implementations are available online, we also included them in our evaluation (see Section 4).

Savary & Piskorski [17] investigated solutions for Polish, also a highly inflectional language. As one subcomponent of their IE platform *SProUT* they filled a gazetteer by "explicitly listing all inflected forms of each entry".

Day & Prukayastha [7] gave an overview of different NER methods especially targeting Indian languages. Their paper presented gazetteer-based and machine learning approaches as well as hybrid solutions.

Al-Jumaily et al. [3] present an NER system for Arabic text mining. They use a token-based approach involving stemming as well as pre- and postfix verification tailored to the Arabic language. Although they aim for real-time applications, they do not give any details about their system's runtime performance.

Al-Rfou & Skiena [4] propose *SpeedRead*, an NER pipeline which they tested to run ten times faster than the *Stanford CoreNLP* pipeline[5]. Unfortunately for us, they only reported runtime performance in terms of tokens per second. In their final results, they say SpeedRead achieves about 153 tokens/sec. Using the word length statistics published by Norvig [14] and assuming an average token length of about five characters, we would end up having 765 char./sec, which is still much too slow for our scenario as we will later see. Even if we assume an average token length of twelve, although more than 90% of all English words are shorter [14], we would still be too slow having 1836 char./sec.

In summary, we found several approaches dealing with either real-time capability or inflection tolerance. One paper even mentioned both, but did not report any concrete speed measures. Nevertheless, doing NER extremely fast is apparently rarely discussed in literature, yet. This may be because usual NER methods operate in only a few seconds, which may be sufficient for many use cases, unfortunately not ours.

We will refer to some of the ideas discussed in this section when presenting our approach in the following.

3 Approach

We focus on the very fast recognition of NEs given as instance labels of ontologies. Moreover, these labels should still be recognized even if they slightly lexically vary as induced by inflection. To achieve this, we exploit knowledge graphs connected or available to our system such as an individual user's Personal Information Model (PIMO) [16] or DBpedia to get more details about the entities, e.g. their specific type. Based on this type, we can then accept different lexical variations per instance according to language information coming from Wiktionary[6], for example. For instance, we should not allow too many variations of person names, whereas we can be more tolerant when dealing with topic, project, organization or location names, especially if they contain adjectives like the *Technical* University of Kaiserslautern or *German* Research Center for *Artificial* Intelligence. As an example, Figure 3 shows all 18 inflected forms of *künstlich* (*artificial*) in German (word `w4` in the figure).

As depicted in Figure 2, we have a hierarchical NE recognizer as the core of our system. It operates on several sub-recognizers, mostly Multi-Layer Finite-State Transducers (MLFST) as described later, each of them having a different focus (configuration). The core recognizer

[5] `https://stanfordnlp.github.io/CoreNLP/`
[6] `https://www.wiktionary.org/`

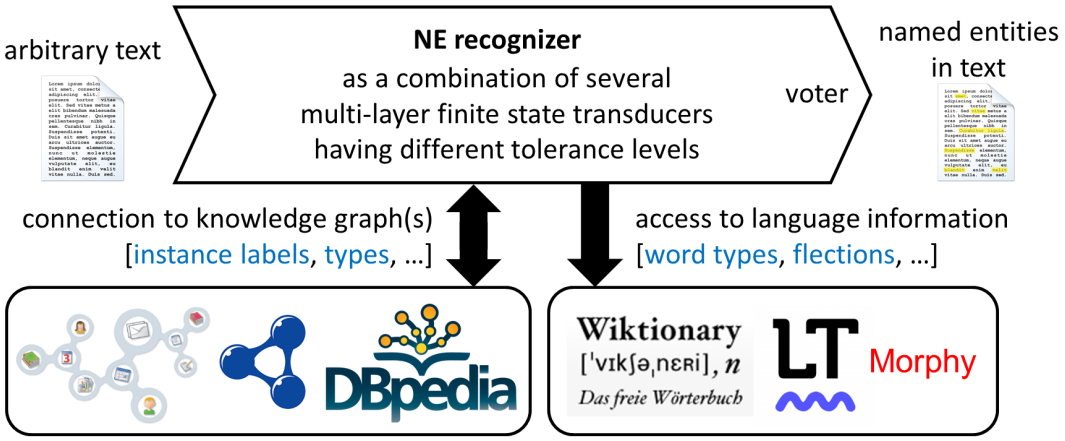


Figure 2 Architecture of our system.

collects their results and decides (votes) which ones to accept. To acquire the entity labels as well as background information, it is connected to knowledge graphs and language information sources as described before. Its individual aspects are discussed in more detail in the following.

3.1 FST-based NER

To meet the aforementioned strict runtime requirements, we basically follow a gazetteer-based approach. The additionally required inflection tolerance is not well compatible with the usually static character of a gazetteer. We thus need enhancements as described in the following.

Our core method is based on the well-known string matching algorithm by Aho & Corasick [2]. It operates on *tries*, i.e. trees whose nodes represent characters, which are traversed synchronously to the processing of each character of the input text. Whenever the traversal ends in an accepting state, there is a string match. Since, in our case, these strings are the labels of NEs, we additionally demand that their ID or URI is returned, which makes the system a Finite-State Transducer (FST). The algorithm basically has linear runtime complexity as discussed later. Our scenario involves a highly dynamic, evolving knowledge graph, in which instances (and especially their labels) can be added, deleted or updated potentially several times per minute. We thus omitted further optimizations like suffix compression in favor of a fast and easy to update FST structure.

3.2 Multi-Layer FST

For runtime performance reasons we decided against sophisticated NLP pipelines (test results and more details in Section 4) and therefore follow the approach of explicitly listing all inflected forms of an entity label as proposed in [17]. Without further ado, this would easily lead to memory performance problems due to a considerable increase of the FST, especially for multi-word terms: The more words such a multi-word term consists of, the more potential combinations exist. Although inflection tolerance is discussed more thoroughly in the paragraph after next, let us just consider a short example here: If we allow each combination of inflected forms of the term *Deutsches Forschungszentrum für Künstliche Intelligenz* (*German Research Center for Artificial Intelligence*, shortly referred to as DFKI), although lots of them are grammatically not correct (as also discussed later), we would end

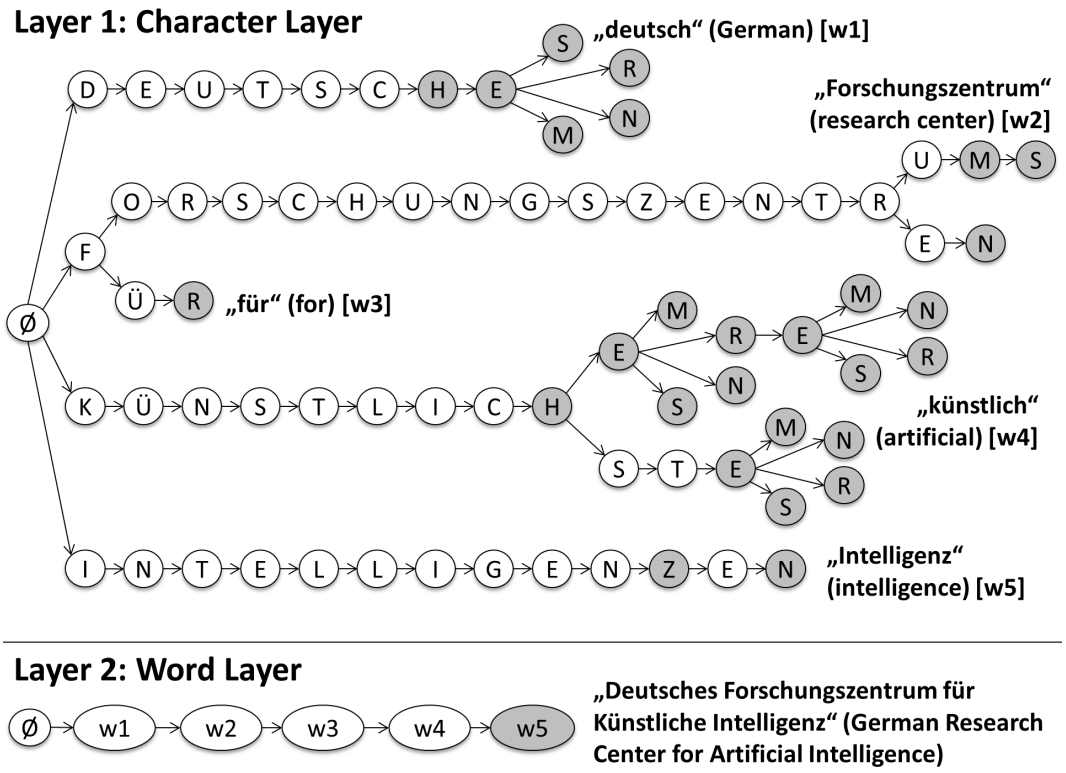


Figure 3 Multi-layer finite transducer consisting of a character and a word layer, and fed with the term *Deutsches Forschungszentrum für Künstliche Intelligenz* (∅: start nodes; w_i: word IDs; gray nodes: accepting states).

up with 576 variations ($= 6 \cdot 3 \cdot 1 \cdot 16 \cdot 2$; see upper part of Figure 3). Inspired by Abney, who proposed the idea of *finite-state cascades* [1], we therefore chose to introduce an additional layer to separate character from word processing, making our system a multi-layer FST as illustrated in Figure 3: Once a word is identified in the first layer (i.e. the FST is in an accepting state; gray node), its ID is passed to the second layer, which checks whether this word may be accepted at this position, either as a single-word or part of a multi-word term. If a term match is detected, its ID/URI is returned. As a consequence, each word and its inflected forms, no matter how often or at which positions (in multi-word terms) they appear, only exist once in the FST, thus preventing it from growing too fast in size.

To avoid backtracking in the word layer, the system processes several options in parallel as shown in Figure 4: Once the character layer recognizes a word, e.g. `w6`, a new word node processor in the second layer is spawned (see upper left part of the figure; purple color). If layer 1 then reports the next word `w7` (highlighted in green), processor 1 goes one step further in the graph now having a traversed path containing both words. Additionally, another processor is spawned, starting directly with `w7`. For this behavior, we use the metaphor of a rake (if you merge all start nodes into a single one, you get the image): Spawning another processor is like adding another tine to the rake. Traversals in the word layer are only possible if the next detected word is a successor of the current one within any term of the FST, which, for example, is not the case when processor 2 tries to handle `w9`, or processor 4 tries to start directly with `w9`. The latter means that there is no term in the

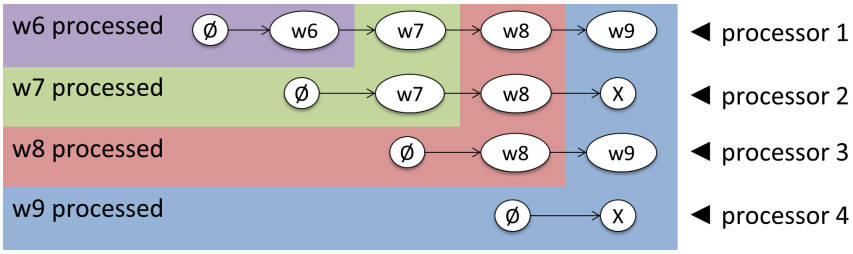


Figure 4 Processing in the word layer: several processors operate in parallel. Their traversed paths are depicted (∅: start nodes; w_i: word IDs; X: failure states).

FST starting with the word `w9`. These two processors are then in a failure state (indicated by "X"). If there was a matching term in their traversed path, it is collected to be later processed by the voter. If that is not the case, the failed processors may be removed from the rake. The second case in which processors are removed, whether they are in a failure state or not, is after an explicit signal from the first layer, e.g. when reaching the end of a file or sentence. Spawning additional processors to evaluate different possibilities in parallel especially originated from the latter. Consider the case of interpreting a dot: It could either indicate the end of a sentence (*"Today, I met my Prof."*), or an abbreviation (*"Prof. Smith was also there."*). Thus, there is a forking in the second layer to evaluate both possibilities separately. In theory, this could lead to endless forking, which is prevented by processors reaching failure states (i.e. given word sequences not matching any term) followed by their removal. The basic steps of our algorithm are given as pseudocode (see Algorithm 1).

3.3 Real-Time Capability

Reading an input text of length n characterwise yields a basic runtime complexity of $\mathcal{O}(n)$. The same is true for processing n characters in the first layer (at most n transitions having a constant amount of operations; no backtracking needed). The processing of a character may lead to the detection of a new word, which then triggers transitions in the word layer. The number of these transitions depends on the number p of processors ("tines in the rake"). p does not depend on n, but on the vocabulary, i.e. all words fed to the FST, especially $w_{\max}$, the maximum number of words in all multi-word terms. Although $p_{\max}$ is constant for a given vocabulary, it may still be very large in worst case[7]. In practical scenarios however, $p \ll p_{\max}$ can be assumed, since the vocabulary is only a tiny fraction of the power set of its words. As a consequence, processors fail very fast due to given word combinations not matching any term in the FST. Considering an additional constant amount of $c > 0$ operations per processor in each transition of the second layer yields an upper limit of $c \cdot p_{\max} \cdot n$. Since n is thus only multiplied with constants, the overall runtime complexity remains $\mathcal{O}(n)$. Although the second layer's overhead is noticeable in practice (as we will see in Section 4), the overall runtime complexity is still linear and benefits our system's applicability in scenarios of real-time processing.

[7] In worst case, a term consisting of $w_{\max}$ words is read, whereas each subterm also exists in the vocabulary. Moreover, for each of these subterms there is an additional variant ending with a dot. This leads to forking after every word and a total amount of $p_{\max} = \sum_{i=1}^{w_{\max}} 2^i$ processors before the first one of them fails and is removed.

Algorithm 1: Basic steps of our MLFST-based NER algorithm in pseudocode.

```
input  : text to process (text)
output : found entities (foundEntities)
foundEntities ← { };
collectedTerms ← { };
c ← first character of text;
w ← c;
while c not equals EOF (end of file or text snippet) do
    if c is whitespace character then
        if w matches in character layer then
            add new word node processor (in word layer);
            for all word node processors p do
                process w with p (may either lead to word match or failure state in p);
            end
        end
        collectedTerms ← collectedTerms ∪ collect matching terms from word layer;
        remove word node processors in failure state (word layer);
        w ← ∅;
    else
        w ← w + c;
    end
    c ← read next character of text (character layer);
end
collectedTerms ← collectedTerms ∪ collect matching terms from word layer;
foundEntities ← do voting on collectedTerms (word layer);
return foundEntities;
```

3.4 Inflection Tolerance

As mentioned before, to accept different lexical variations of terms, e.g. induced by inflection, we utilize information coming from connected ontologies as well as other language information sources. Concerning the latter, we use a lemmatization table extracted from *LanguageTool*[8], an open source proofreading software for several languages, which itself contains binary files of *Morfologik* to look up part-of-speech data. Such entries look as follows:

```
künstlich     künstlich   ADJ:PRD:GRU
künstliche    künstlich   ADJ:AKK:PLU:FEM:GRU:SOL
künstlichem   künstlich   ADJ:DAT:SIN:MAS:GRU:SOL
...
```

They contain the inflected form, its lemma as well as declension information like word class, case, number, gender, etc. We additionally used *Wiktionary*, a free wiki-based dictionary, whose data[9] we extracted using *DKpro JWKTL*[10] [18]. Nevertheless, there were still lots of words not covered by any of these sources, especially compound words like *Forschungszentrum*

[8] `https://github.com/languagetool-org` (uses `https://github.com/morfologik`)
[9] `https://dumps.wikimedia.org/` (dump file of 2016-07-01)
[10] `https://dkpro.github.io/dkpro-jwktl/`

(*research center*). To counter this, we additionally implemented heuristics like longest suffix matching to decompound words and use the inflected forms of the last part (if available). In the case of *Forschungszentrum* not being in our database, the heuristic would first look for the word *orschungszentrum* (fails), then *rschungszentrum* (fails), *schungszentrum* (fails), etc., until finally finding *zentrum* and using its inflected forms, i.e. *Zentrum*, *Zentrums* and *Zentren*. The matching part of the original word is then replaced with these inflected forms as shown in Figure 3. The heuristic additionally expects a parameter indicating the minimum length of the remaining suffix (e.g. five characters) to receive more meaningful results. Our tool is thus able to handle yet unknown words to a certain extent without user interaction. In this regard, let us revisit the aforementioned 576 variations of the term DFKI. As also mentioned, most of them are grammatically not correct. Since we also want to handle yet unknown words, especially compound ones, while keeping the user interaction as low as possible (not asking for feedback), we decided to accept all variations obtained as the Cartesian product of all inflected forms of each of a term's words. We assume that grammatically wrong variants do rarely occur in given texts and if they do, users will agree with the entity being recognized despite the misspelling. Nevertheless, the question remains whether this decision considerably increases the false positive rate. We will address this in Section 4. To avoid actually harmful false positives of incorrectly inflected variants, we exploit additional ontological information like the type of an entity. For example, the name of a person tolerates far less variants than the name of a topic. Basically, we only allow a possessive/genitive case "s" at the end, like stated before. As a consequence, our NE recognizer is actually not just a single MLFST, but a combination of several ones each having a different configuration. Currently, there is one having higher and another one having lower tolerance. The latter, for example, contains person names. There is also an option to especially deal with acronyms: They do not only require exact matches, moreover all characters need to be uppercase. To further avoid non-meaningful variants, we only use adjective and noun information from the lemmatization table, which reduces ambiguities when not having thorough NLP information. This is a compromise we can accept, since labels more often contain nouns and adjectives than verbs.

When processing input text, the different MLFST operate in parallel. In the end, a voter receives, assesses, filters and finally returns their results. Additionally, each MLFST has its own internal voter which assesses all results simultaneously present in a processing rake. In the current implementation, these voters follow a strategy of only keeping the longest match, e.g. if the term *personal information management* is found in the text, the also matching terms of *personal information* and *information management* would be discarded.

4 Evaluation

4.1 Setting

Besides finding out how fast our NE recognizer performs in practice, we were especially interested whether our design decisions (see Section 3) would lead to a considerable increase in false positives. We were thus looking for large amounts of German natural language texts (prose) written by different people to test our approach. The German Wikipedia meets this requirement but lacks ground truth data for the inflected forms present in these texts. We therefore decided to only look at the wikilinks (see Figure 1, top section, blue words) and take them as a silver standard: A human has annotated terms in the text (often in inflected form) with the label of their respective Wikipedia article (typically in basic form). Figure 1 also shows that users themselves decide which terms they annotate: There are lots of entities

(highlighted in green), which are not annotated although there is a Wikipedia article for them. This is especially true for self-references, e.g. the term *Aussagenlogik* is not annotated in "its own" article (i.e. the one about *Aussagenlogik*). Recognizers fed with such terms, would nevertheless find them, which has to be considered when measuring precision.

Regardless of possible shortcuts, annotations are structured as follows: the term appearing in the text and the name of the Wikipedia article it refers to (in the following also shortly called *the link*) are written in double brackets separated by a pipe symbol, e.g. `[[Häuser|Haus]]` (plural form of *house* appears in the text, whereas the article is labeled with the singular form). Since inflection usually just changes one to four characters, the Levenshtein distance (LD) between term and link can help us identifying samples we could use to evaluate our approach. Note that independent term-link-combinations like `[hometown|Eton]` or adjective-noun-combinations like `[entscheidbar|Entscheidbarkeit]`(*decidable/decidability*) are undesirably also covered by such an LD-based heuristic. On the other hand, this evaluation approach offers millions of inflection samples (we ran our tests on 3.9M articles having 50.4M annotations).

We downloaded German Wikipedia dump files[11] and used 3.9M article names as a basis for feeding our recognizers. Disambiguation information in brackets like in *"Berlin (Russland)"* (a village in Russia sharing its name with the German capital) were removed (this raises disambiguation issues as discussed later). We also removed number-, symbol- and single-character-only labels, since they were not relevant for our investigations. As ontological background information we used types[12] coming from DBpedia, which were available for about 0.5M entities. For types like person, city, film, etc., we applied a low tolerance strategy (i.e. possessive/genitive case "s" is the only accepted variation), whereas all other ones were treated with higher tolerance.

4.2 Evaluated Named Entity Recognizers

We evaluated our MLFST approach against three baseline methods. The first and most obvious one, *StemFST*, was also implemented by us and uses the MLFST's character layer combined with the *Lucene*[13] *German Stemmer*, which is based on [5]. The other methods are the previously mentioned ones by Dlugolinsky et al. [8], who made two of their gazetteers available online[14]: one based on hash-map multi-way trees (*HMT*), and the other based on first child-next sibling binary trees (*CST*). Both produce the same results in terms of found NEs, but differ in memory consumption and runtime performance.

After filtering and editing as mentioned in the previous paragraph, we had slightly above 3.3M article names of the German Wikipedia that we fed to all four NE recognizers. HMT and CST take these terms without further changes. StemFST splits each term into words and reassembles it after stemming them. Then it adds the altered term to its FST. MLFST does the same but instead of stemming the words, it looks up (or tries to infer) their inflected forms. Completely filled, the inner high-tolerance MLFST contained 8.5M character nodes and 3.5M word nodes, the low-tolerance part kept 1M and 0.4M nodes, respectively.

[11] `https://dumps.wikimedia.org/` (dump file of 2016-11-01)
[12] `https://downloads.dbpedia.org/3.9/de/instance_types_de.ttl.bz2`
[13] `https://lucene.apache.org/`
[14] `http://ikt.ui.sav.sk/gazetteer/`

4.3 Results

All computations were performed on a notebook having an Intel Core i7-4910MQ 2.9 GHz CPU and 16 GB RAM, running on Windows 7 (64-bit).

HMT only needed 10.4 min for processing 3.9M articles (9.4B characters), while the others needed 31.0 to 47.7 min (see Figure 5). Figure 6 shows that HMT trades memory efficiency for speed, since it is the only recognizer passing the 1 GiB mark by needing 3.5 GiB. The others needed 0.72 to 0.96 GiB.

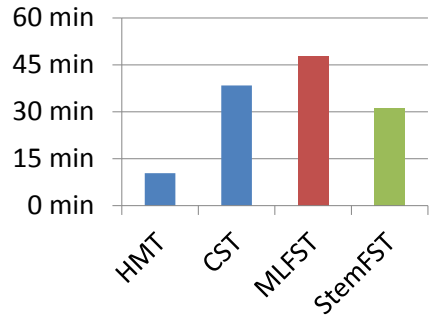


Figure 5 Processing time.

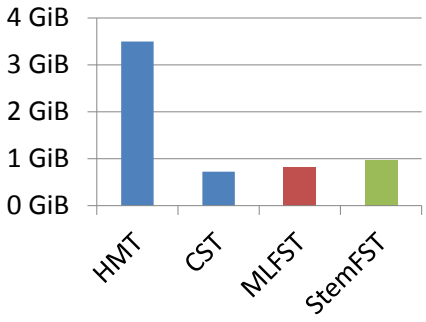


Figure 6 Memory usage.

4.3.1 Recall

Let us next consider recall: All recognizers reached values slightly below or above 70%. Figure 8 additionally shows the results itemized by LD. If term and link match exactly (`LD=0`, which is the case for 69% of all annotations), all recognition rates are above 92%[15]. In LD ranges of `LD=1` to `LD=4` (11% of all annotations), HMT/CST's recall is close to 0%, whereas MLFST still has rates of 79%, 66%, 36% and 8%, respectively. StemFST even has slightly higher rates. Reaching recall near 100% should not be expected, since not all variations are caused by inflection and their number decreases with increasing LD. For higher LD values (`LD>4`, 21% of all annotations), all recognition rates are close to 0%.

4.3.2 Precision

Concerning precision, we already mentioned the problem of how to measure it adequately. We decided to calculate multiple values: P_O measures precision only for terms *overlapping* with annotation positions, because only there we have "ground truth" data. As shown in Figure 7, some found terms (purple highlighting) are not exactly matching the actual annotation (blue word, highlighted in green as the only true positive). If terms are overlapping with the annotation, we count them as a false positive. P_A counts *all* terms not exactly

Figure 7 Example sentence to illustrate the different precision values.

[15] errors in the dump and imperfect parsing caused a slight decrease (100% expected)

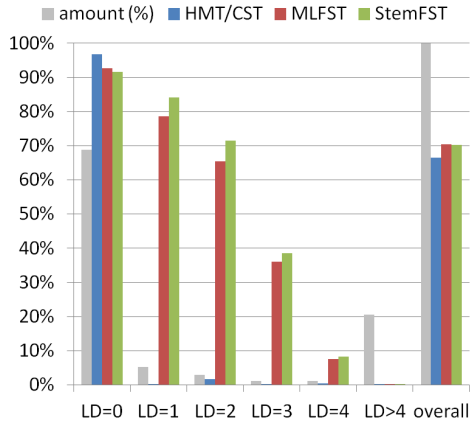


Figure 8 Recall itemized by Levenshtein distance of term and link.

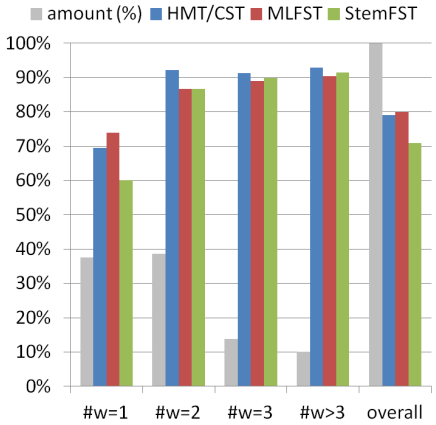


Figure 9 Precision P_O^* itemized by the terms' number of words (#w).

matching as false positives, especially also the non-overlapping ones (red highlighting). Since disambiguation was out of this paper's scope and there are labels belonging to more than 1000 instances (e.g. *Jewish cemetery*), it makes a large difference whether or not we additionally count more than 1000 false positives for each true positive in a text. We thus introduce P_O^* and P_A^*, which count multiple entities having the same label only once. P_O^* is 79% for HMT/CST and 80% for MLFST, while StemFST only reaches 71%. Figure 9 additionally depicts P_O^* itemized by the number of words a term consists of. For multi-word terms, all approaches achieve values between 87% and 92%. There is a remarkable difference for single word terms: Here, stemming seems to be too rough causing terms to lose their specifity and StemFST to lose 14% compared to MLFST, which performs best having 74%. The other overall precision values P_O, P_A and P_A^* are shown in Figure 10. They are far lower than P_O^* due to the aforementioned reasons. However, in a short experiment, in which students annotated some randomly chosen articles manually, we observed values for P_A^* that were similar to P_O^* above. We thus have a slight indication that P_A^* (depicted above) heavily underestimates our algorithm's precision. Finally answering one of our initial research questions: the false positive rate of MLFST is not considerably higher (in some cases even lower) than with the other recognizers.

4.3.3 Runtime Performance

Regarding runtime performance, MLFST and StemFST process between 3281 and 5048 characters per millisecond and are thus comparable to CST as illustrated in Figure 11. HMT is about three times faster at the expense of memory consumption (see Figure 6). All tested recognizers are by orders of magnitude faster than basic NLP pipelines. We tested OpenNLP[16] and CoreNLP using a basic pipeline consisting only of a tokenizer, sentence splitter and part-of-speech tagger. Although no NER-specific analyzers like noun chunkers or type classifiers were added yet, their processing time was already out of our targeted range. Running the basic pipeline on all 3.9M articles would presumably have taken about 18 days in the case of CoreNLP, for example.

[16] `https://opennlp.apache.org/`

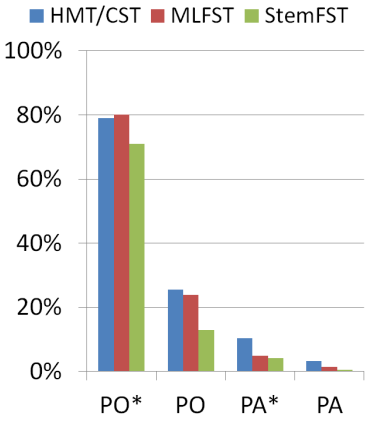


Figure 10 Precision: P^*_O, P_O, P^*_A, P_A.

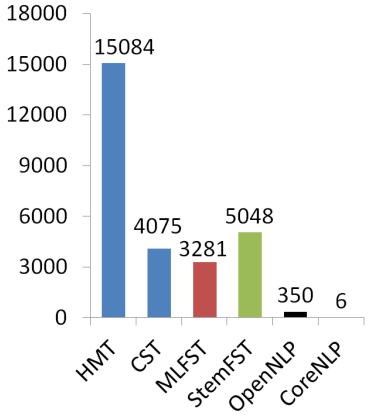


Figure 11 Processed characters per ms.

5 Conclusion & Outlook

In this paper, we presented an ontology-based NER approach that is comparably fast as available high speed methods while outperforming them in the recognition of terms that lexically vary slightly, e.g. induced by inflection. We were thus able to narrow the quality gap to more sophisticated but also much slower NLP pipelines a bit more without losing real-time capable runtime performance.

In the future, we plan to additionally incorporate StemFST into MLFST, since its recall was slightly better for multi-word terms. Additionally, we could add more layers scanning for patterns like phrases that indicate todos or appointments, Hearst patterns [9], etc. There is also much potential for improving the language capabilities of our approach, e.g. improved rules and heuristics (e.g. to infer inflections) or multi-language support. Last but not least, we plan to incorporate disambiguation mechanisms by exploiting the explicated user context available in our system.

References

1 Steven Abney. Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, 1996.

2 Alfred V. Aho and Margaret J. Corasick. Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, 18(6):333–340, 1975.

3 Harith Al-Jumaily, Paloma Martínez, José L. Martínez-Fernández, and Erik Van der Goot. A real time Named Entity Recognition system for Arabic text mining. *Language Resources and Evaluation*, 46(4):543–563, 2012.

4 Rami Al-Rfou and Steven Skiena. SpeedRead: A Fast Named Entity Recognition Pipeline. *Proceedings 24th International Conference on Computational Linguistics (COLING 2012)*, pages 51–66, 2012.

5 Jörg Caumanns. A fast and simple stemming algorithm for German words. Technical Report TR B 99-16, Center für Digitale Systeme, Freie Universität Berlin, 1999.

6 Hamish Cunningham, Valentin Tablan, Angus Roberts, and Kalina Bontcheva. Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. *PLoS computational biology*, 9(2):e1002854, 2013.

7 Arindam Dey and Bipul Syam Prukayastha. Named Entity Recognition using Gazetteer Method and N-gram Technique for an Inflectional Language: A Hybrid Approach. *International Journal of Computer Applications*, 84(9), 2013.

8 Stefan Dlugolinskỳ, Giang Nguyen, Michal Laclavík, and Martin Šeleng. Character gazetteer for Named Entity Recognition with linear matching complexity. In *3rd World Congress on Information and Communication Technologies (WICT)*, pages 361–365. IEEE, 2013.

9 Marti A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Vol. 2*, pages 539–545. Association for Computational Linguistics, 1992.

10 Christian Jilek, Markus Schröder, Sven Schwarz, Heiko Maus, and Andreas Dengel. Context Spaces as the Cornerstone of a Near-Transparent and Self-Reorganizing Semantic Desktop. In *The Semantic Web: ESWC 2018 Satellite Events*, pages 89–94. Springer, 2018.

11 Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, pages 1–8. ACM, 2011.

12 Giang Nguyen, Štefan Dlugolinskỳ, Michal Laclavík, Martin Šeleng, and Viet Tran. Next Improvement Towards Linear Named Entity Recognition Using Character Gazetteers. In *Advanced Computational Methods for Knowledge Engineering*, pages 255–265. Springer, 2014.

13 Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann, 1993.

14 Peter Norvig. English Letter Frequency Counts: Mayzner Revisited or ETAOIN SRHLDCU, 2013. accessed: 2018-08-18. URL: `http://norvig.com/mayzner.html`.

15 Leo Sauermann, Ansgar Bernardi, and Andreas Dengel. Overview and Outlook on the Semantic Desktop. In *Proceedings of the 1st Workshop on the Semantic Desktop at the ISWC 2005 Conference*, pages 74–91. CEUR-WS, 2005.

16 Leo Sauermann, Ludger van Elst, and Andreas Dengel. PIMO – a framework for representing personal information models. In *Proceedings of I-Media '07 and I-Semantics '07*, pages 270–277. Know-Center, Austria, 2007.

17 Agata Savary and Jakub Piskorski. Lexicons and grammars for named entity annotation in the National corpus of Polish. In *18th International Conference Intelligent Information Systems*, pages 141–154, 2010.

18 Torsten Zesch, Christof Müller, and Iryna Gurevych. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, pages 1646–1652, 2008.

Validation Methodology for Expert-Annotated Datasets: Event Annotation Case Study

Oana Inel[1]
Delft University of Technology, The Netherlands
Vrije Universiteit Amsterdam, The Netherlands
o.inel@tudelft.nl, oana.inel@vu.nl

Lora Aroyo
Google Research, New York, US
loraa@google.com

Abstract

Event detection is still a difficult task due to the complexity and the ambiguity of such entities. On the one hand, we observe a low inter-annotator agreement among experts when annotating events, disregarding the multitude of existing annotation guidelines and their numerous revisions. On the other hand, event extraction systems have a lower measured performance in terms of F1-score compared to other types of entities such as people or locations. In this paper we study the consistency and completeness of expert-annotated datasets for events and time expressions. We propose a data-agnostic validation methodology of such datasets in terms of consistency and completeness. Furthermore, we combine the power of crowds and machines to correct and extend expert-annotated datasets of events. We show the benefit of using crowd-annotated events to train and evaluate a state-of-the-art event extraction system. Our results show that the crowd-annotated events increase the performance of the system by at least 5.3%.

2012 ACM Subject Classification Information systems → Crowdsourcing; Human-centered computing → Empirical studies in HCI; Computing methodologies → Machine learning

Keywords and phrases Crowdsourcing, Human-in-the-Loop, Event Extraction, Time Extraction

Digital Object Identifier 10.4230/OASIcs.LDK.2019.12

1 Introduction

Natural language processing (NLP) tasks span a large variety of applications [14], such as event extraction, temporal expressions extraction, named entity recognition, among others. While the performance of named entity recognition tools is constantly improving, the event extraction performance is still poor. On the one hand, events are vague and can have multiple perspectives, interpretations and granularities [16]. On the other hand, there is hardly a single, standardized way to represent events. Instead, we find a plethora of annotation guidelines, standards and datasets created, adapted and extended by human experts [33]. Although the annotation guidelines are aimed to ease the annotation task, the inter-annotator agreement values reported are still low, ranging between 0.78 and 0.87 [7, 33]. Current research [7, 33, 15] acknowledges the fact that expert-annotated datasets could be inconsistently annotated or could contain ambiguous labels, but there is no standardized way of measuring if they indeed contain inconsistent or incomplete annotations.

In the natural language processing field, crowdsourcing is extensively used as a mean of gathering fast and reliable annotations [29]. Although, typically, crowd annotations are evaluated against experts annotations by means of majority vote approaches, more recent

[1] Corresponding author


© Oana Inel and Lora Aroyo;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 12; pp. 12:1–12:15

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

approaches focus on capturing the *inter-annotator disagreement* [1] and the creation of ambiguity-aware crowd-annotated datasets [12].

In this paper we present a data-agnostic validation methodology for expert annotated datasets. We investigate the degree of consistency and completeness of expert-annotated datasets and we propose an ambiguity-aware crowdsourcing approach to validate, correct and improve them. We apply this methodology on the expert annotated datasets of events and time expressions, namely TempEval-3 Gold (Gold) and TempEval-3 Platinum (Platinum), which were used in the TempEval-3 Time Annotation[2] task at SemEval 2013. To show the added value of employing crowd workers for providing event annotations, we use the crowd-annotated events to train and evaluate a state-of-the-art event extraction system which participated in the challenge. Therefore, we investigate the following research questions:

RQ1: *How reliable are expert-annotated datasets in terms of consistency and completeness?*

RQ2: *Can we improve the reliability of expert-annotated datasets in terms of consistency and completeness through crowdsourcing?*

To answer these research questions we make the following contributions:

- data-agnostic validation methodology of expert-annotated datasets in terms of consistency and completeness;
- 4,202 crowd-annotated English sentences from the TempEval-3 Gold and TempEval-3 Platinum datasets with events and 121 crowd-annotated sentences from the TempEval-3 Platinum dataset with time expressions;
- training and evaluation of a state-of-the-art system for event extraction with ambiguity-aware crowd-driven event annotations.

We make available the crowdsourcing annotation templates for all experiments, the scripts used for our validation methodology and the crowdsourcing results in the project repository[3].

The remainder of this paper is structured as follows. Section 2 reviews related work in the field of event extraction by focusing on automatic, crowdsourcing and human-in-the-loop approaches. Section 3 describes the dataset and Section 4 introduces our data-agnostic validation methodology. Section 5 presents the results of our data-agnostic validation methodology for measuring the consistency and completeness of expert-annotated datasets. Section 6 presents and discusses the results of our crowdsourcing experiments and the learning outcomes. Finally, Section 7 draws conclusions and introduces future work.

2 Related Work

We review related work on event and time expression detection in three main areas: automatic approaches (Section 2.1), crowdsourcing approaches (Section 2.2) and hybrid, human-in-the-loop approaches (Section 2.3). We focus on the identification of linguistic mentions of type event and time expression, as opposed to identifying named entities of type event and time.

2.1 Automatic Approaches

We review event and time expression detection systems that use domain-agnostic expert-annotated datasets for training and evaluation, such as datasets following the TimeML [26] specifications. This category includes the TempEval-3 dataset, that we use in the current research. We only focus on the detection of events and time expressions, without looking into event classification, time expression normalization or the relations between the two.

[2] https://www.cs.york.ac.uk/semeval-2013/task1/index.html
[3] https://github.com/CrowdTruth/Event-Extraction

For event extraction the majority of the participating systems in the TempEval-3 Time Annotation task used a supervised, knowledge-driven approach with various types of classifiers such as Conditional Random Fields (JUCSE) [20], Maximum Entropy (ATT and NavyTime) [18, 9] and Logistic Regression (ClearTK and KUL) [2, 19] and features such as morphological, semantic, lexical, among others. The TIPSem system [23], the best performing system in the previous challenge from the same series, outperformed all the participants with an F1-score of 82.89 compared to 81.05 of the ATT1 [18] system on identifying the event mention. To the best of our knowledge, the TIPSem [23] system and the CRF4TimeML [6] system (F1-score of 81.87) are currently the best performing systems trained on TimeML datasets.

For temporal expression extraction the best performance in terms of F1-score was 90.32, exhibited by both the NavyTime [9] and SUTime [10] systems. However, they both used a rule-based approach without actually making use of the training data. The next best performing systems on temporal expression extraction, with F1-scores above 0.90, were HeidelTime [31] and ClearTK [2], both using only expert-annotated data as training.

All the aforementioned systems have been evaluated on the TempEval-3 Platinum dataset, an expert-annotated corpus [32]. Although potential ambiguity and errors have been identified in this dataset in previous research [6, 33], the dataset has not been revised. As opposed to this approach, we also evaluate the performance of the ClearTK [2] system with ambiguity-aware crowd-driven event mentions.

2.2 Crowdsourcing Approaches

Crowdsourcing proved to be a reliable approach to gather large amounts of labeled data for many natural language processing tasks such as temporal event ordering [29], causal relation identification between events [5], event factuality [21], event validity [8], among others. As researched [1] showed, disagreement in crowdsourced annotations can be an indication of ambiguity, ambiguous classes of polysemy for event nominals were identified in [30] and ambiguous frames in [12]. In [7], the authors present a crowdsourcing approach for identifying events and time expressions in English and Italian sentences by asking the crowd to highlight phrases in the sentence that refer to events or time. A different approach was taken in [21], where the crowd had to validate one event, at a time, in a sentence. In all the aforementioned approaches, the annotations of the crowd were evaluated against expert annotations.

In this research we combine and extend the approaches proposed in [7] and [21] by asking the crowd to validate in each sentence a set of potential events and time expressions and highlight the missing ones. Moreover, before running the main crowdsourcing study, we run extensive small scale pilot experiments to identify the optimal crowdsourcing settings. Since events and, in a smaller proportion, time expressions are highly ambiguous mentions, we follow and apply the CrowdTruth disagreement-aware methodology [1], similarly to [12], to aggregate and evaluate the crowd annotations. These annotations are then evaluated against expert and also machine annotations. Furthermore, we use the crowd-annotated events as both training and evaluation data for a state-of-the-art event extraction system from the TempEval-3 challenge, namely ClearTK [2].

2.3 Hybrid and Human-in-the-loop Approaches

In NLP, hybrid human-machine approaches have been mainly envisioned on named entity extraction and typing [15] and named entity extraction and linking [11]. The human-machine hybrid NER system published in [3] focused on decomposing individual examples into either examples that can be labelled by automatic tools or by the crowd. Hybrid approaches for

event and temporal expression extraction also focused on combining various machine learning approaches with human rules [25]. Although active learning approaches have been used for building event or temporal expression extraction systems [4, 22], the labels are still gathered by means of expert annotators instead of crowdsourcing. In [21], however, the authors use the crowd labels for training a supervised event extraction system.

Current hybrid approaches for event extraction focus on a predefined set of event types, while our approach is suitable for general events. Similarly to [21], we use the crowd-labelled events to train an existing state-of-the-art system for event extraction on the TempEval-3 corpus, but also to evaluate it.

3 Dataset

We focus our analysis on expert-annotated entities of type event and time expression in the TempEval-3 Gold (Gold) and TempEval-3 Platinum (Platinum) datasets from the SemEval 2013 task called TempEval-3 Time Annotation. The Platinum dataset was used to test the performance of the participating systems and the Gold dataset was used for the development of the systems. A detailed description of these two datasets can be found in [27, 28, 32].

We used the TimeML-CAT-Converter[4] and Stanford CoreNLP [24] to split the documents into sentences and tokens and to annotate the tokens with part-of-speech (POS[5]) tags and lemmas. In Table 1 we show the overview of the Gold (G) and Platinum (P) datasets (DS), *i.e.*, the number of documents, sentences, tokens, events and time expressions (times). The Gold dataset contains 256 documents which were split into 3,953 sentences and around 100k tokens and the Platinum dataset contains 20 documents, 273 sentences and around 7k tokens. The Gold dataset contains 3,604 events and 1,450 times, while the Platinum dataset contains 746 events and 138 times, and thus, 3.07 events and 1.27 times per sentence, on average.

Table 1 Overview of TempEval-3 - Gold (G) and TempEval-3 Platinum (P) Datasets (DS).

DS	# Doc	# Sent	# Tokens	#Ann Sent Events	#Ann Sent Times	# Events	# Times	Avg. #Events per Sent	Avg. #Times per Sent
G	256	3,953	≈ 100k	3,604	1,464	11,129	1,822	3.08	1.24
P	20	273	≈ 7k	243	106	746	138	3.06	1.30

Events and Times POS Tags Distribution: Similarly to [33], we looked at the POS tag distribution of events and time expressions in the Gold and Platinum datasets. In both datasets the majority of the events annotated are either verbs or nouns. Adjectives, adverbs and, in a smaller proportion, prepositions are also annotated as events. The Platinum dataset also contains 3 multi-token events composed of numerals. Regarding time expressions, around half of the annotated ones are composed of multiple tokens with various POS tags such as nouns, numbers, preposition, adverbs and adjectives.

Events and Times Tokens and Lemmas: Table 2 shows the number of distinct event and time tokens and lemmas by considering as well their POS tags. On average, in the Gold dataset an event token appears 3.86 times (between 1 and 993 times, *i.e.*, the token "said") while an event lemma appears around 5.94 times (between 1 and 1,154 times, *i.e.*, the lemma "say"). In the Platinum dataset an event token appears on average 1.38 times and an event lemma around 1.69 times. Regarding time expressions, tokens and lemmas appear on average 2.89 times in the Gold dataset and around 1.46 times in the Platinum dataset.

[4] https://github.com/paramitamirza/TimeML-CAT-Converter
[5] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Table 2 Overview of Distinct Event and Time Tokens and Lemmas.

DS	Events		Times	
	Distinct Tokens	Distinct Lemmas	Distinct Tokens	Distinct Lemmas
Gold	2,883	1,871	630	623
Platinum	537	440	94	94

Sentences without Event and Time Annotations: As shown in Table 1, a fraction of the total amount of sentences contained in the two datasets do not contain annotated events, *i.e.*, 349 in Gold and 30 in Platinum, or time expressions, *i.e.*, 2,489 in Gold and 167 in Platinum.

4 Experimental Methodology

In this section we describe our data-agnostic validation methodology of expert-annotated datasets in terms of consistency and completeness. The goal of our experimental methodology is two-fold: *(1)* to measure the reliability of expert-annotated datasets for events and time expressions in terms of consistency and completeness and *(2)* to define an optimal crowdsourcing annotation template to improve the reliability of expert-annotated datasets for events and time expressions in terms of consistency and completeness. The two research questions defined in Section 1 and the following hypotheses guide our experimental methodology:

H1.1 (consistency): Tokens are annotated with different types across datasets.
H1.2 (consistency): Annotation guidelines for events are not used consistently.
H1.3 (completeness): Occurrences of the same previously annotated event tokens or time expression tokens are not annotated by experts.
H1.4 (completeness): Occurrences of the same previously annotated event lemmas or time expression lemmas are not annotated by experts.
H2.1 (reliability): Asking the crowd annotators to motivate their answer increases the reliability of their annotations.
H2.2 (reliability): Gathering event annotations from a large pool of crowd workers provides reliable results in terms of F1-score when compared to expert annotators.
H2.3 (reliability): Crowd-driven event annotations are a reliable way of improving the consistency and completeness of expert-annotated event datasets.

The first step of our methodology, described in Section 4.1, is guided by and extends previously published work on consistency and completeness analysis of expert-annotated datasets of named entities (location, organization, person and role) [15], of events in the TempEval-3 Gold, PropBank/NomBank and FactBank datasets [33] and of events and time expressions in all TempEval-3 datasets [6]. The second step of our methodology adapts the crowdsourcing approach proposed in [15] to improve, complete and correct expert-annotated datasets of events and time expressions. We derive the optimal crowdsourcing annotation template by experimenting with different annotation template independent variables, as described in Section 4.2. Finally, we train and evaluate the ClearTK [2] state-of-the-art event extraction system with crowd-annotated events, as described in Section 4.3.

4.1 Ground Truth Consistency and Completeness

We test hypotheses **H1.1-4** by performing a headroom measurement on the consistency and completeness of expert-annotated entities of type event and time in the TempEval-3 Gold and TempEval-3 Platinum datasets. For consistency (**H1.1-2**) we *(1)* check whether an entity span is annotated with different types across datasets and *(2)* review the experts' adherence to the annotation guidelines. For completeness (**H1.3-4**) we *(1)* verify for each

event and time expression token and lemma the proportion in which it was annotated as an event or as a time expression and *(2)* inspect the sentences without annotated events or time expressions to verify whether they might contain missed mentions.

Table 3 Overview of Performed Pilot (P1 to P8) and Main (M1 & M2) Crowdsourcing Experiments.

Exp.	Input Data				Crowdsourcing Template	
	#Sent	Entity Type	DS	Entity Values	Annotation Guidelines	Annotation Value
P1	50	Event Time	P	Experts (P) & Tools	Explicit Definition	Entities
P2	50	Event Time	P	Experts (P) & Tools	Explicit Definition	Entities + Motivation (NONE)
P3	50	Event Time	P	Experts (P) & Tools	Explicit Definition	Entities + Motivation (ALL)
P4	50	Event Time	P	Experts (P) & Tools	Implicit Definition	Entities
P5	50	Event Time	P	Experts (P) & Tools	Implicit Definition	Entities + Motivation (NONE)
P6	50	Event Time	P	Experts (P) & Tools	Implicit Definition	Entities + Motivation (ALL)
P7	50	Event Time	P	Experts (G&P) & Tools & Missing	Explicit Definition	Entities + Motivation (ALL)
P8	50	Event Time	P	Experts (G&P) & Tools & Missing	Explicit Definition	Entities + Motivation (ALL) + Highlight
M1	4,202	Event	G&P	Experts (G&P) & Tools & Missing	Explicit Definition	Events + Motivation (ALL) + Highlight
M2	121	Time	G&P	Experts (G&P) & Tools & Missing	Explicit Definition	Times + Motivation (ALL) + Highlight

4.2 Crowdsourcing Experiments

We further test **H1.3-4** through a series of pilot crowdsourcing experiments aiming to improve the ground truth datasets for events and time expressions. We start with a set of 16 *pilot experiments* (eight experiments for event annotation and eight for time expression annotation), P1 to P8 rows as shown in Table 3, in which we experiment with the input data that the crowd is requested to annotate and the design of the crowdsourcing template, similarly to [17]. The role of these pilot experiments is to *obtain the optimal annotation template design*, following **H2.1-2**. We run these experiments on the Figure Eight[6] platform, using level 2 workers from English-speaking countries, *i.e.*, UK, US, CAN and AUS, for each annotation we pay ¢3 (for annotation value without highlight functionality) or ¢4 (for annotation value with highlight functionality) and we ask 20 workers to annotate each sentence.

For each pilot experiment we used 50 sentences from the TempEval-3 Platinum (P) dataset as input data. The crowd needs to validate or add, through highlight, entities of type event or time expression. We vary the list of entities that the crowd needs to validate as follows. In the first six pilot experiments (P1-P6 in Table 3) the crowd was asked to validate only the entities annotated by the experts and returned by the systems participating in the

[6] `https://www.figure-eight.com`

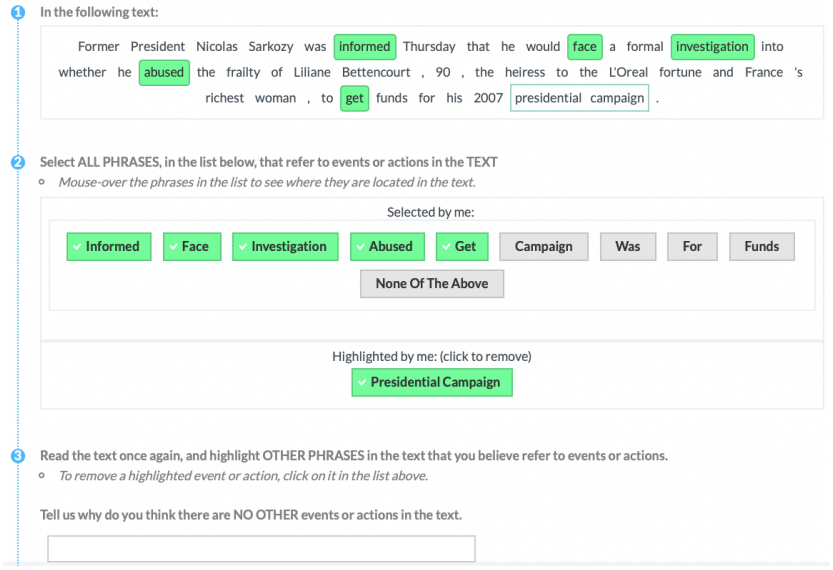


Figure 1 Screenshot of the Main Crowdsourcing Template (M1) to Validate and Highlight Events.

TempEval-3 task. In P7-P8, we expanded the list of entities to be validated with potentially missing entities such as *(1)* annotated entities in the Gold (G) and Platinum (P) datasets and *(2)* any other entity that was annotated in other sentence, but not in the current one.

As part of the crowdsourcing template design we experiment with the annotation guidelines and the annotation values. We request annotators to validate mentions that are both explicit (phrases that refer to events or actions, or temporal expressions) and implicit (phrases that refer to things happening in the past, present, or future, or that involve times, dates, durations, periods, etc.). For the annotation value, we experiment with four options: *(1)* validation of event or time entities, *(2)* validation of those entities with motivation (only when there is no valid entity), *(3)* validation of those entities with motivation (regardless of whether there are valid entities) and *(4)* validation of entities with motivation (regardless of whether there are valid entities) and highlight of potential missed entities.

Main Experiments. We evaluate the outcome of the pilot experiments against the expert annotations to derive the optimal crowdsourcing template in terms of performance (F1-score) to validate, correct and improve datasets for events and time expressions. We run the *main crowdsourcing experiments* on the entire dataset, with the optimal setup. The main crowdsourcing experiments (M1 an M2, the last two rows in Table 3) have the following setup: the input data consists of sentences and events or time expressions annotated by experts, participating systems in the TempEval-3 task and potentially missed events or time expressions; the crowdsourcing template uses explicit definitions and validation of entities with motivation (regardless of whether there are valid entities) and highlight of missed entities. Figure 1 shows the design of the crowdsourcing template for events. We run the *main experiments* on the Figure Eight platform, using level 2 workers from English-speaking countries. Each sentence is annotated by 15 workers and for each annotation we pay ¢4.

4.2.1 Crowd Annotation Aggregation

We aggregate and evaluate the crowd annotations using the CrowdTruth approach for open-ended tasks [13, 12]. First, we define the *worker vector*, *i.e.*, the decision of a worker over an input unit, *i.e.*, a sentence. The worker vector in our case is composed of all entities (either events or time expressions) to be validated or have been highlighted for a given sentence and the value *"none"* (capturing cases when there are no valid entities). Each component in the worker vector gets a value of 1 if the worker selected the entity as valid and 0, otherwise. The sum of all *worker vectors* for a given sentence results in the *sentence vector*. The worker and sentence vectors are then used to compute the following ambiguity-aware metrics:

- *entity-sentence score* ***(ESS)***: expresses the likelihood of each entity e (event of time expression) to be valid for the given sentence s; ESS is computed as the ratio of workers that picked the entity as valid over all the workers that annotated the sentence, weighted by the worker quality; the higher the ESS value, the more clear e is expressed in s;
- *sentence quality score* ***(SQS)***: expresses the workers agreement over one sentence s; SQS is computed as the average cosine similarity of all worker vectors for a sentence s, weighted by the worker quality and entity quality;
- *worker quality score* ***(WQS)***: expresses the overall agreement of one worker with the rest of the workers; WQS is computed using cosine similarity metrics, weighted by the sentence quality and entity quality;
- *entity quality score* ***(EQS)***: being an open-ended task, $EQS = 1$.

These ambiguity-aware metrics are mutually dependent (*i.e.*, they are computed in an iterative dynamic fashion), which means that each aforementioned quality metric depends on the values of the other two metrics. Thus, low quality workers can not decrease the quality of the sentences, and low quality sentences can not decrease the quality of the workers.

4.3 Training & Evaluating the ClearTK Event Extraction System

We used the crowd-annotated events to train and evaluate the ClearTK[7] [2] event extraction system reviewed in Section 2.1, that participated in the TempEval-3 challenge. The selection of the system was made purely based on the availability of the code to easily retrain and evaluate the models. ClearTK [2] uses BIO token chunking for event identification, using the following features: token text, stem, part-of-speech, the syntactic category of the token's parent in the constituency tree, the text of the first sibling of the token in the constituency tree and the preceding and following 3 tokens.

First, after gathering the crowd annotations for both the Gold and Platinum datasets, we apply the aggregation and evaluation metrics presented in Section 4.2.1. Second, we create multiple development (from Gold documents) and evaluation (from Platinum datasets) sets by splitting the crowd-annotated events based on their entity-sentence score, *i.e.*, for every entity-sentence score threshold between 0 and 1, with a step of 0.05. Therefore, we obtain 20 sets of development and evaluation datasets, each containing all the events with a score higher than the respective threshold. Finally, we perform the following four types of experiments to test hypothesis **H2.3**:

- train the system on expert-annotated events and test it on expert-annotated events
- train the system on expert-annotated events and test it on crowd-annotated events
- train the system on crowd-annotated events and test it on expert-annotated events
- train the system on crowd-annotated events and test it on crowd-annotated events

[7] `https://github.com/ClearTK/cleartk`

For all the aforementioned experiments we did not fine-tuned the model's parameters, but we used the ones that performed the best in the TempEval-3 event-extent extraction task.

5 Consistency and Completeness of Expert Annotations

In this section we inspect the consistency and completeness of expert-annotated event and time expression mentions in the TempEval-3 Gold and Platinum datasets, following the hypotheses **H1.1-4**. First, we measure the consistency of the expert-annotated mentions regarding the span of the mentions, the type of the annotated mentions and the adherence to the annotation guidelines in Section 5.1. Second, we measure the completeness of the expert-annotated events and times at the level of part-of-speech distribution and tokens and lemmas and we analyze the sentences without annotated events in Section 5.2.

5.1 Consistency of Expert Annotations

The events annotated by experts in the TempEval-3 Gold (Gold) dataset consist of a single token. Even when the event refers to a multi-token named event, such as "World War II" or "Hurricane Hugo", the experts only mark as event a single token, such as "war" or "hurricane". Interestingly, the TempEval-3 Platinum (Platinum) dataset contains multi-token events composed of numerals, such as "$ 250", "400 million". These events are *not consistent with the latest annotation guidelines* [28] (**H1.2**), since the events of type numeral should be removed. An inconsistency identified in [6] shows that the Platinum dataset contains the noun "season" annotated as event once, while in other sentences from the Gold dataset, it is annotated as a time expression. Furthermore, we observe that the token "tenure" is annotated as an event in the Gold dataset and as a time expression in the Platinum dataset. Therefore, besides a *mention type inconsistency*, we also see an *inconsistency across the training and the evaluation datasets*, proving **H1.1**. Another observation that we made is that overlapping mentions of both type event and time expression are not possible. For example, the word "election" was annotated as event in Platinum dataset, but in the Gold dataset is treated as a time expression, in the word phrase "election day".

5.2 Completeness of Expert Annotations

The completeness analysis follows the setup published in [33]. In the current research, we build on top of this analysis and extend it on a new dataset – TempEval-3 Platinum – and on a new entity type – time expression. Furthermore, we provide entity completeness statistics on the sentences without expert annotated events.

5.2.1 POS Tags Distribution

We analyze the distribution of POS tags (as returned by Stanford CoreNLP) across the events and times annotated by experts in the TempEval-3 Gold and Platinum datasets. For the events annotated by experts in the Platinum dataset, we see consistent observations with the ones published in Table 3 in [33]. Overall, in both datasets *verbs have the highest coverage as events* (63.29% in Gold and 54.43% in Platinum). However, there is still a *significant number of verbs that were not annotated as events*, such as the verbs "participate" or "follow". The *nouns annotated as events have a much lower coverage* (7.89% in Gold and 8.62% in Platinum). Interestingly, in the *Platinum dataset, the rate of verbs annotated as events is lower compared to the Gold dataset, but the rate of nouns annotated as events*

is higher than the Gold dataset. Since, on average, not more than 1% of the total amount of adjectives, adverbs and prepositions were annotated as events by the experts in both datasets, we assume they might introduce ambiguity.

In both datasets, around 50% of all the annotated time expressions consists of single tokens of POS noun, numeral, adjective and adverb. While the rate of nouns and numerals annotated as times in the Platinum dataset is almost equal, in the Gold dataset, there are around 4 times more nouns annotated as time expressions compared to numerals. All the multi-token time expressions are combinations of tokens having at least a noun or a numeral.

5.2.2 Tokens and Lemmas

Table 4 presents the overview of the potential inconsistencies encountered in the expert-annotated events in the Platinum dataset, by looking at event tokens and lemmas across all (*ALL*) POS tags and per individual POS tag. As in the analysis performed in [33], we identify possible inconsistencies at the token level - *not all instances of an event are always annotated as events* (*e.g.*, the noun "apology" is annotated as event in 1 out of 6 cases, the verb "keep" is annotated as event in 1 case out of 9). This type of inconsistency appears for 74 distinct event tokens out of a total of 537 distinct event token - POS tag pairs (*i.e.*, 13.85% cases). Similarly, we also identify inconsistencies at the lemma level - *not all lemma instances of an event are always annotated as events* (*e.g.*, the noun "charge" is annotated as event in 1 out of 5 lemma-based occurrences, the verb "say" is annotated as event in 63 cases out of 65). There are 90 such distinct lemma-based inconsistency cases out of 440 unique pairs event lemma - POS tag (*i.e.*, 20.59% cases). The amount of *inconsistencies at the level of event lemma is higher than at the level of event token*, which means that only certain lemmas of a token are usually annotated as events by experts. Overall, the least amount of disagreement is seen for events that are either verbs or nouns.

Table 4 Event Inconsistencies at the Level of Event Tokens and Lemmas in TempEval-3 Platinum.

	Total Inconsistencies (%)		**Distinct Inconsistencies (%)**	
	Token	Lemma	Token	Lemma
ALL	287 (27.86%)	476 (39.04%)	74 (13.85%)	90 (20.59%)
VB	215 (28.25%)	388 (41.54%)	42 (11.26%)	53 (18.79%)
NN	66 (27.61%)	82 (32.15%)	27 (19.56%)	32 (24.24%)
JJ	5 (19.23%)	5 (19.23%)	4 (20.0%)	4 (20.0%)
RB	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)

Regarding time expressions, we observed that in the Platinum dataset year mentions such as "1953", "2010" are not annotated as time expressions by experts. Further, we looked into the multi-token time expressions and computed how many times a mention was missed. In the Platinum dataset, we found only two missed mentions, both at the level of token and lemma, while in the Gold dataset we found 91 missed mentions at the token level and 105 mentions at the lemma level. Overall, 46 time expression mentions were not always annotated out of 497 unique time expression tokens and 492 time expressions lemmas.

5.2.3 Sentences without Annotated Events

In Figure 2 and Figure 3 we plotted for each sentence without annotated events (in the TempEval-3 Gold dataset and respectively, in the TempEval-3 Platinum dataset) on the first *y axis* the number of tokens in each sentence (ordered) and on the second *y axis (1)* the total number of verb POS tags contained in the sentence and *(2)* the total number of event lemmas

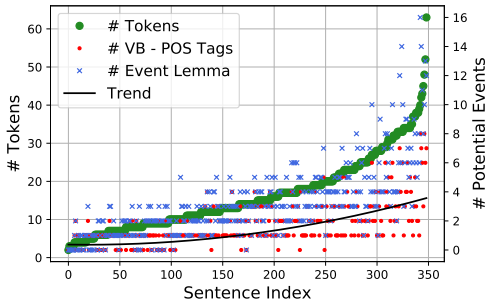


Figure 2 Overview of Potentially Missed Events in Sentences from the TempEval-3 Gold Dataset without Expert Event Annotations.

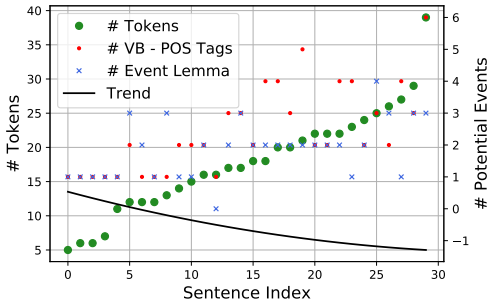


Figure 3 Overview of Potentially Missed Events in Sentences from the TempEval-3 Platinum Dataset without Expert Event Annotations.

that were annotated in other sentences, but not the current one. We observe a positive correlation between the number of verb POS tags contained in the sentences and the number of annotated event lemmas in other sentences, which means that many of the verbs in these sentences were actually tagged as events in other sentences. Even though the correlation does not seem as strong for the sentences in the TempEval-3 Platinum dataset (Figure 3, we believe this is due to the low number of sentences. Therefore, based on these observations and the ones presented in the previous subsections, we re-emphasize the incompleteness in the expert annotations, closely correlated to our hypotheses **H1.3-4**.

6 Results

In this section we report on the results[8] of the pilot and main *crowdsourcing experiments* in Section 6.1 and the results of employing the crowd-annotated events to train and evaluate an event extraction system in Section 6.2.

6.1 Crowdsourcing Experiments

In the 16 *crowdsourcing pilot experiments* we gathered in total 8,000 crowd annotations from a total of 134 unique workers. The total cost of these pilots was equal to $624. We start by evaluating the performance of the crowd in terms of precision (P), recall (R) and F1-score, in comparison with the expert annotations, in each pilot experiment. In Table 5 we see the overview of this analysis. To compare the crowd annotations with the expert annotations, we first applied the crowd aggregation metrics introduced in Section 4.2.1. As a result, each entity (either event or time expression) validated by the crowd gets an entity-sentence score (ESS) with values between 0 and 1, which shows the likelihood of that entity to be valid. First of all, we observe that the crowd performs better when they are provided with explicit definitions of the entities that they need to validate (see results for P1, P2, P3). Second, in alignment with our **H2.1** hypothesis and confirming it, we observe that when the crowd is asked to motivate their answers, their performance is improved (see results for P3 and P6).

As described in Section 4.2, in P7 and P8 we increased considerably the list of entities to be validated by the crowd. Furthermore, in P8 we also gave them the option to highlight

[8] `https://github.com/CrowdTruth/Event-Extraction`

Table 5 Crowd vs. Experts Performance Comparison on all Crowdsourcing Pilot Experiments.

	Events					Time Expressions				
	Thresh	P	R	F1-score	#TP	Thresh	P	R	F1-score	#TP
P1	0.35	0.84	0.93	0.89	152	0.60	0.71	0.86	0.78	50
P2	0.15	0.79	1.0	0.88	164	0.50	0.67	0.86	0.75	50
P3	0.50	0.83	0.98	**0.90**	161	0.60	0.76	0.84	**0.80**	49
P4	0.40	0.84	0.95	0.89	154	0.65	0.73	0.82	0.78	48
P5	0.35	0.80	0.98	0.88	159	0.65	0.80	0.72	0.76	42
P6	0.45	0.84	0.95	**0.89**	157	0.60	0.79	0.81	**0.80**	47
P7	0.45	0.75	0.95	0.84	156	0.65	0.75	0.83	0.78	48
P8	0.50	0.73	0.93	0.83	155	0.75	0.78	0.77	0.78	45

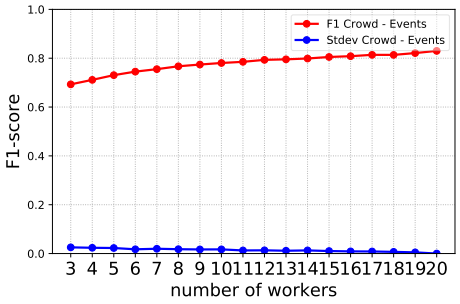


Figure 4 Events Crowd F1-score at the Best ESS Threshold for Various # Workers.

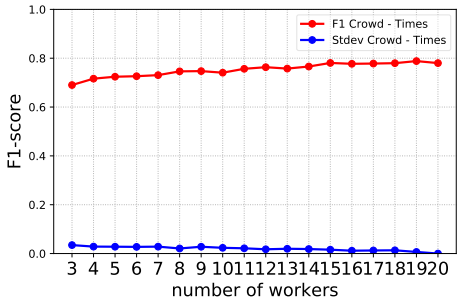


Figure 5 Times Crowd F1-score at the Best ESS Threshold for Various # Workers.

potentially missing entities, *i.e.*, entities that are not found in the validation list. However, the crowd still performs well when compared to the experts. Even though the overall F1-score slightly dropped, the total number of true positive entities remains almost the same. The drop in F1-score is due to the fact that the crowd finds more relevant entities than the ones annotated by experts. Thus, we hypothesize that this is a viable and reliable way of gathering missing entities and correct the expert inconsistencies. Therefore, based on these observations, we ran the *main experiment* using the P8 setup.

Next, we focused on understanding what would be the optimal number of crowd annotations needed per sentence, at the best performing *ESS* threshold for the crowd. For each number of workers between 3 and 20, we averaged their F1-score for a total of 100 runs, by randomly generating sets of [3:20] workers. In Figure 4 and Figure 5 we plot both the average F1-score and the standard deviation (stdev) among all the runs for the pilot experiment P8, for events and respectively, time expressions. In both cases, we observe that around 15 workers the F1-score of the crowd stabilizes and the stdev is negligible. Furthermore, this observation aligns with our **H2.2** hypothesis which says that enough annotations from the crowd provides reliable results when compared to experts.

In the *main experiments* we gathered 63,030 crowd annotations from 160 unique workers and the total cost of the experiments was $3,112, by running the setup of P8 with 15 workers, on the entire set of sentences. In order to see how the crowd compares to the expert annotations, we again performed the evaluation of the crowd entities for every entity-sentence score threshold. Thus, for time expressions we got the best performing F1-score of 0.70 at thresholds between [0.65 and 0.90] and for events we got the best performing F1-score of

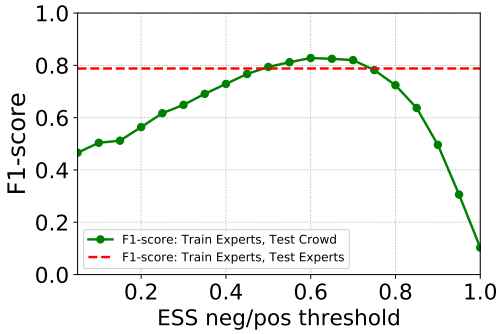


Figure 6 ClearTK F1-score when Trained on Expert Events and Tested on Crowd Events.

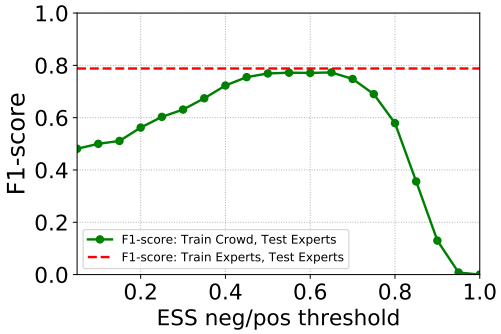


Figure 7 ClearTK F1-score when Trained on Crowd Events and Tested on Expert Events.

0.81 at a threshold of 0.60. Overall, we see that these results are consistent with the ones in the pilot experiments, even though the scale is much larger. Therefore, we acknowledge that the crowd is able to provide *consistent event and time expression annotations*.

Table 6 ClearTK F1-score when Trained on Crowd Events and Tested with Crowd Events.

Crowd ESS Threshold		Test								
		0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70
Train	0.30	**0.824**	**0.806**	0.783	0.75	0.721	0.697	0.669	0.649	0.623
	0.35	0.797	**0.798**	**0.798**	0.786	0.764	0.744	0.72	0.699	0.674
	0.40	0.766	0.783	**0.797**	**0.799**	0.791	0.778	0.765	0.745	0.72
	0.45	0.738	0.769	0.797	**0.818**	**0.823**	0.81	0.802	0.79	0.768
	0.50	0.71	0.747	0.779	0.814	**0.828**	**0.827**	**0.829**	0.815	0.796
	0.55	0.687	0.727	0.761	0.799	0.821	**0.826**	**0.83**	0.819	0.804
	0.60	0.658	0.698	0.735	0.776	0.802	0.816	**0.826**	0.824	0.819
	0.65	0.639	0.681	0.721	0.764	0.79	0.807	0.820	**0.822**	0.819
	0.70	0.596	0.638	0.673	0.716	0.747	0.771	0.791	0.800	**0.805**

6.2 Training and Evaluating with Crowd Events

We report on the results of the ClearTK event extraction systems, when trained and evaluated on crowd-annotated events. It is important to acknowledge that for training purposes we used the systems' parameters that performed the best in the TempEval-3 task, and we did not fine-tuned them to better fit our training data.

In Figure 6 we plotted the F1-score of the system when trained on expert events and evaluated on crowd events, for every event-sentence score (ESS) threshold. We can observe that between the ESS thresholds [0.5:0.75] the system performs much better than when it is evaluated on the expert events. The measured F1-score of the ClearTK system in the TempEval-3 task was 0.788, while the maximum achieved F1-score when evaluated on crowd events reaches values of around 0.83. However, when we train the system on crowd events and we test it on expert events, the performance achieved by the system is only almost as good (0.77) as the reported F1-score of 0.788. This happens due to the fact that the crowd annotates events in a more consistent way, while experts, according to Section 5, are missing potentially valid annotations. Finally, in Table 6 we show the results of both training and

evaluating the ClearTK system on crowd events, for each ESS threshold between [0.30:0.70]. The results clearly indicate that the crowd event annotations are a reliable and consistent way of providing event annotations (correlated to **H2.3**) - the crowd performs the best when trained and evaluated at similar ESS thresholds. Furthermore, we observe that while for training the best performing threshold could vary between [0.50:0.60], for testing the threshold of 0.60 seems to provide the best and most consistent F1-scores, up to 0.830.

7 Conclusion and Future Work

In this paper we proposed a data-agnostic validation methodology for expert-annotated datasets and we showed its application on the case of events and, to some extent, time expressions. We propose a set of analytics to measure the consistency and completeness of such datasets and a crowdsourcing approach to mitigate these problems. We conducted extensive pilot crowdsourcing experiments and we derived the optimal setup to gather event and time expression annotations based on them. We showed that the crowd-annotated events are a reliable dataset to train and evaluate state-of-the-art event extraction systems. Furthermore, we showed that the performance of such systems can be improved by at least 5.3% when both trained and evaluated on crowd data.

As part of future work we plan to use the crowd-annotated events for *(1)* training and evaluating a larger range of state-of-the-art event extraction systems, as well as *(2)* running more extensive experiments such as fine-tunning the learning parameters based on the crowd-training data and using different crowd event thresholds. Furthermore, we plan to investigate the impact that ambiguous events have in training and evaluating event extraction tools. Finally, we plan to replicate the experiment with time expressions and investigate the added value of gathering crowd annotations for this mention type.

References

1 L. Aroyo and C. Welty. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24, 2015.

2 S. Bethard. ClearTK-TimeML: A minimalist approach to TempEval 2013. In ** SEM, Volume 2: SemEval 2013*, volume 2, pages 10–14, 2013.

3 K. Braunschweig, M. Thiele, J. Eberius, and W. Lehner. Enhancing named entity extraction by effectively incorporating the crowd. *BTW Workshop*, 2013.

4 K. Cao, X. Li, M. Fan, and R. Grishman. Improving event detection with active learning. In *International Conference Recent Advances in Natural Language Processing*, pages 72–77, 2015.

5 T. Caselli and O. Inel. Crowdsourcing StoryLines: Harnessing the Crowd for Causal Relation Annotation. In *Proceedings of the Workshop Events and Stories in the News*, 2018.

6 T. Caselli and R. Morante. Systems' Agreements and Disagreements in Temporal Processing: An Extensive Error Analysis of the TempEval-3 Task. In *LREC*, 2018.

7 T. Caselli, R. Sprugnoli, and O. Inel. Temporal Information Annotation: Crowd vs. Experts. In *LREC*, 2016.

8 A. Ceroni, U. Gadiraju, and M. Fisichella. Justevents: A crowdsourced corpus for event validation with strict temporal constraints. In *ECIR*, pages 484–492, 2017.

9 N. Chambers. NavyTime: Event and time ordering from raw text. Technical report, Naval Academy Annapolis MD, 2013.

10 A. Chang and C. D. Manning. SUTime: Evaluation in tempeval-3. In ** SEM, Volume 2: SemEval 2013*, volume 2, pages 78–82, 2013.

11 G. Demartini. Hybrid human–machine information systems: Challenges and opportunities. *Computer Networks*, 90:5–13, 2015.

12 A. Dumitrache, L. Aroyo, and C. Welty. Capturing Ambiguity in Crowdsourcing Frame Disambiguation. In *HCOMP 2018*, pages 12–20, 2018.

13 A. Dumitrache, O. Inel, L. Aroyo, B. Timmermans, and C. Welty. CrowdTruth 2.0: Quality Metrics for Crowdsourcing with Disagreement. *arXiv preprint arXiv:1808.06080*, 2018.

14 A. Gangemi. A comparison of knowledge extraction tools for the semantic web. In *ESWC Conference*, pages 351–366, 2013.

15 O. Inel and L. Aroyo. Harnessing diversity in crowds and machines for better NER performance. In *European Semantic Web Conference*, pages 289–304, 2017.

16 O. Inel, L. Aroyo, C. Welty, and R.-J. Sips. Domain-independent quality measures for crowd truth disagreement. *DeRiVE Workshop*, page 2, 2013.

17 O. Inel, G. Haralabopoulos, D. Li, C. Van Gysel, Z. Szlávik, E. Simperl, E. Kanoulas, and L. Aroyo. Studying Topical Relevance with Evidence-based Crowdsourcing. In *CIKM*, pages 1253–1262. ACM, 2018.

18 H. Jung and A. Stent. ATT1: Temporal annotation using big windows and rich syntactic and semantic features. In **SEM, Volume 2: SemEval 2013*, volume 2, pages 20–24, 2013.

19 O. Kolomiyets and M.-F. Moens. KUL: Data-driven approach to temporal parsing of newswire articles. In ** SEM, Volume 2: SemEval 2013*, volume 2, pages 83–87, 2013.

20 A. K. Kolya, A. Kundu, R. Gupta, A. Ekbal, and S. Bandyopadhyay. JU_CSE: A CRF based approach to annotation of temporal expression, event and temporal relations. In **SEM, Volume 2: SemEval 2013*, volume 2, 2013.

21 K. Lee, Y. Artzi, Y. Choi, and L. Zettlemoyer. Event detection and factuality assessment with non-expert supervision. In *EMNLP*, pages 1643–1648, 2015.

22 S. Liao and R. Grishman. Using prediction from sentential scope to build a pseudo co-testing learner for event extraction. In *IJCNLP*, pages 714–722, 2011.

23 H. Llorens, E. Saquete, and B. Navarro. TIPsem (English and Spanish): Evaluating CRFs and semantic roles in TempEval-2. In *SemEval*, 2010.

24 C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics System Demonstrations*, pages 55–60, 2014.

25 C. Min, M. Srikanth, and A. Fowler. LCC-TE: a hybrid approach to temporal relation identification in news text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 219–222, 2007.

26 J. Pustejovsky, R. Knippen, J. Littman, and R. Saurí. Temporal and event information in natural language text. *Language resources and evaluation*, 39(2):123–164, 2005.

27 J. Pustejovsky, J. Littman, R. Saurí, and M. Verhagen. TimeBank 1.2. *Linguistic Data Consortium*, 40, 2006.

28 R. Saurí, J. Littman, B. Knippen, R. Gaizauskas, A. Setzer, and J. Pustejovsky. TimeML annotation guidelines. *Version*, 1(1):31, 2006.

29 R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*, pages 254–263, 2008.

30 R. Sprugnoli and A. Lenci. Crowdsourcing for the identification of event nominals: an experiment. In *LREC*, pages 1949–1955, 2014.

31 J. Strötgen, J. Zell, and M. Gertz. Heideltime: Tuning english and developing spanish resources for tempeval-3. In ** SEM, Volume 2: SemEval 2013*, volume 2, pages 15–19, 2013.

32 N. UzZaman, H. Llorens, L. Derczynski, J. Allen, M. Verhagen, and J. Pustejovsky. SemEval-2013 Task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In ** SEM, Volume 2: SemEval 2013*, pages 1–9, 2013.

33 C. Van Son, O. Inel, R. Morante, L. Aroyo, and P. Vossen. Resource Interoperability for Sustainable Benchmarking: The Case of Events. In *LREC*, 2018.

A Proposal for a Two-Way Journey on Validating Locations in Unstructured and Structured Data

Ilkcan Keles
Aalborg University, Dept. of Computer Science, Denmark
ilkcan@cs.aau.dk

Omar Qawasmeh
Univ. Lyon, CNRS, Lab. Hubert Curien UMR 5516, F-42023 Saint-Étienne, France
omar.alqawasmeh@univ-st-etienne.fr

Tabea Tietz
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
Karlsruhe Institute of Technology, Germany
tabea.tietz@fiz-karlsruhe.de

Ludovica Marinucci
Semantic Technology Laboratory (STLab), Istituto di Scienze e Tecnologie della Cognizione-Consiglio Nazionale delle Ricerche (ISTC-CNR), Rome, Italy
ludovica.marinucci@istc.cnr.it

Roberto Reda
Department of Computer Science and Engineering, University of Bologna, Italy
roberto.reda@unibo.it

Marieke van Erp
KNAW Humanities Cluster, DHLab, The Netherlands
marieke.van.erp@dh.huc.knaw.nl

Abstract

The Web of Data has grown explosively over the past few years, and as with any dataset, there are bound to be invalid statements in the data, as well as gaps. Natural Language Processing (NLP) is gaining interest to fill gaps in data by transforming (unstructured) text into structured data. However, there is currently a fundamental mismatch in approaches between Linked Data and NLP as the latter is often based on statistical methods, and the former on explicitly modelling knowledge. However, these fields can strengthen each other by joining forces. In this position paper, we argue that using linked data to validate the output of an NLP system, and using textual data to validate Linked Open Data (LOD) cloud statements is a promising research avenue. We illustrate our proposal with a proof of concept on a corpus of historical travel stories.

2012 ACM Subject Classification Computing methodologies → Natural language processing

Keywords and phrases data validity, natural language processing, linked data

Digital Object Identifier 10.4230/OASIcs.LDK.2019.13

Category Short Paper

Acknowledgements This work was made possible by the *International Semantic Web Research Summer School* in Bertinoro, July 2018. The authors would like to thank the Summer School directors, Valentina Presutti and Harald Sack, as well as the tutors, the organizing team and the fellow students, in particular Amanda Pacini de Moura, Amr Azzam and Amina Annane for their suggestions and input.

© Ilkcan Keles, Omar Qawasmeh, Tabea Tietz, Ludovica Marinucci, Roberto Reda, and Marieke van Erp;
licensed under Creative Commons License CC-BY

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 13; pp. 13:1–13:8

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Even today, most of the content on the Web is available only in unstructured format, and in natural language text in particular. As large volumes of non-electronic textual documents, such as books and manuscripts in libraries and archives, are being digitised, undergoing optical character recognition (OCR) and made available online [12], we are presented with a huge potential of unstructured data that could feed the growth of the Linked Data Cloud.[1]

To integrate this content into the Web of Data, we need effective and efficient techniques to extract and capture the relevant data [5]. Natural Language Processing (NLP) encompasses a variety of computational techniques for the automatic analysis and representation of human language. As such, NLP can arguably be used to produce structured datasets from unstructured textual documents, which in turn could be used to enrich, compare and/or match with existing Linked Data sets. However, NLP systems are not without errors, and neither is Linked Data. We therefore need to ensure that information contained in structured datasets is valid.

This raises two main issues for data validity: **Textual Data Validity**, defined as the validity of information contained in texts, and **Linked Data validity**, defined as the validity of information contained in structured datasets, e.g. DBpedia or GeoNames. Textual data validity corresponds to the case whether one is not sure regarding whether the text contains correct or up-to-date information. Texts are not always written to be updated, for example a travel diary of a person provides his/her experiences during a specific time period using the information valid at that time. Unless particularly interested in providing a travel guide for future travellers, authors often do not return to their original text to add updates. For example, the updated location names remained unchanged in the text. By connecting information in such a publication to more recently updated information, such as a gazetteer that contains information on changes of location names, we can find out the place the author mentions in the text. To illustrate, if the text contains the name of 'Monte San Giuliano', we can infer that it corresponds to the contemporary location named 'Erice'.[2] On the other hand, linked data validity corresponds to the case where the validity of the structured datasets is under question since not all structured datasets contain correct information. For this reason, by connecting a dataset to a text, for example to the original source material, statements in a database can be checked with respect to the information provided by the text. A schematic overview of this process is presented in Figure 1.

We propose that structured data extracted from text through NLP is a fruitful approach to address both issues, depending on the case at hand: structured data from reliable sources could be used to validate data extracted with NLP, and reliable textual sources could be processed with NLP techniques to be used as a reference knowledge base to validate Linked Data sets. This leads us to our definition of validity that covers both cases from an NLP perspective: We assess the data element as valid

- whenever an entity is extracted from a text and refers to an entity in a trusted Linked Data dataset and the entity's properties extracted from text are aligned with the trusted dataset, or
- when an entity is present in a structured dataset, refers to an entity described in a trusted text and the entity's properties are aligned with the information extracted from the trusted text.

[1] Linked Open Data Cloud `http://lod-cloud.net/` Last retrieved 10 January 2019
[2] `https://en.wikipedia.org/wiki/Erice` Last retrieved: 10 January 2019

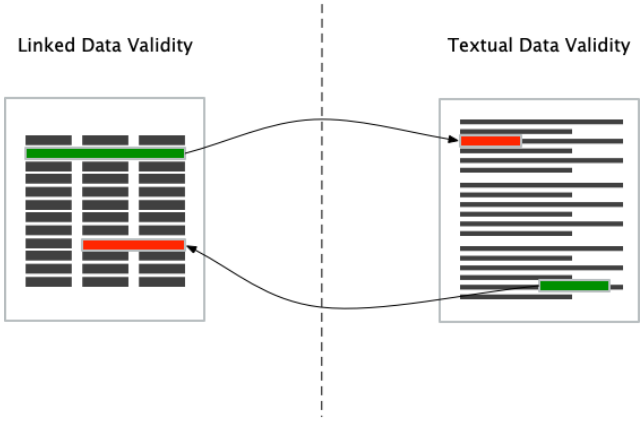


Figure 1 Interplay between Linked Data Validity and Textual Data Validity where Linked Data can be used to validate information contained in text, and information contained in text can be used to validate information contained in Linked Data.

Trust in this sense refers to metadata quality (e.g. precision and recall) as well as intrinsic data qualities [1].

In order to demonstrate this, we performed an analysis on a corpus of Italian travel writings by native English speakers[3] to extract data on locations, and then matched the extracted data with the two structured open data sets on geographic locations.

The remainder of this paper is structured as follows: Section 2 presents related work. Section 3 presents the use case description, highlighting the issues with the current disconnect between linked data and text. Section 4 concludes this work.

2 Related Work

Our proposed approach relies on using external knowledge bases in order to validate the quality of locations' named entities in historical travel writings, thus placing it in the realm of entity linking [7]. Whilst entity linking can cover a variety of entity types, we focus on location linking, which presents a host of problems specific to the geographical information systems domain.

Existing approaches for identifying which location names refer to which localities are summarized in [11]. The article describes the positional uncertainties and extent of vagueness frequently associated with the place names and with the differences between common users perception and the representation of places in gazetteers. The article focuses on approaches from the search/information retrieval domain, which often cannot benefit from potentially rich background information that linked data sources can provide.

A venture into location linking using semantic web resources is presented in [10]. In this paper, Van Erp et al. propose an automatic approach for georeferencing textual localities identified in a database of animal specimens using GeoNames,[4] Google Maps and the Global Biodiversity Information Facility (GBIF) [8].

[3] `https://sites.google.com/view/travelwritingsonitaly/` Last retrieved 10 January 2019
[4] `https://geonames.org` Last retrieved 10 January 2019

An approach for historical entity linking is presented in [3]. Two use cases are presented:
1. Histpop: the Online Historical Population Reports for Britain and Ireland (1801 to 1937) and
2. BOPCRIS: the Journals of the House of Lords (1688 to 1854).

A ranking system to validate the extracted places by taking advantage of GeoNames and Wikipedia is presented. However, the authors do not make any assumptions about whether the data in GeoNames or the sources from which they extract information is valid or not.

Ceolin et al. [2] propose an approach to address the uncertainty of categorical Web data. They used Beta-Binomial, Dirichlet-Multinomial and Dirichlet Process models in order to handle the validity issue. The authors focus on two validity issues, which are the validity of multi-authoring (i.e. the nature of the web data) and the time variability. In this paper, we address the general validity without focusing on the possible sources of invalidity.

3 Use case: Historical Travel Writings

In this section, we describe our use case through a corpus of historical travel writings which we try to validate against several widely used knowledge bases.

3.1 Resource

We have chosen to work with a corpus of historical writings regarding travel itineraries named as 'Two days we have passed with the ancients... Visions of Italy between XIX and XX century' [9].[5] We propose that this dataset provides rich use cases for addressing the textual data validity defined in Section 1.

1. It contains 57 books that correspond to the accounts written by travelers who are native English speakers traveling in Italy.
2. The corpus consists of the accounts of travelers who have visited Italy within the period of 1867 and 1932. These writings share a common genre, namely 'travel writing'. Therefore, we expect to extract location entities that are valid during the time of the travelling. However, given that the corpus covers a span of 75 years, it potentially includes cases of contradicting information due to various updates on geographical entities.
3. The corpus might also contain missing or invalid information due to the fact that the travelers included in the dataset are not Italian natives, and therefore we cannot assume that they are experts on the places they visited.
4. The corpus also contains pieces of non-factual data, such as the travelers' opinions and impressions.

To validate the locations from the travel writings corpus, we chose structured data sources that deal with geographical entities: GeoNames[4] and DBpedia.[6] GeoNames is a database of geographical names that describes more than 11 million location entities. The project was initiated by geographical information retrieval researchers. The core database is provided by official government sources and users are able to update and improve the database by manually editing its information. Ambassadors from all continents contribute to the GeoNames dataset with their specific expertise.

[5] Italian Travel Writings Corpus `https://sites.google.com/view/travelwritingsonitaly/` Last retrieved 10 January 2019
[6] `https://dbpedia.org`

In addition to a dedicated geographical dataset, we selected DBpedia, the structured database based on Wikipedia, the crowdsourced encyclopaedia. The current version of DBpedia contains around 735,000 places. Information in DBpedia is not updated live, but around twice a year, thus, it is not sensitive to live information, e.g. an earthquake in a certain location or a sudden political conflict between states. However, since working with historical data in this case study and not with live events, we pose that it is reasonable to include geographical information from DBpedia. An added feature of DBpedia over Geonames is that it contains more contextual information about a location which may help the validation process.

3.2 Approach

Textual data validity is difficult to separate from the information extraction process from text, as in that process often background resources are also used. However, to validate an extracted piece of information from text, we propose that deeper background knowledge is used than is customary. Many approaches such as DBpedia spotlight [6] utilize some information from the Wikipedia abstract as well as general information on the knowledge resource. Ideally, multiple resources are used, as well as domain-specific resources and reasoning over the domain, as laid out in [4].

Linked Data validity refers to the validation of Linked Data. To identify whether a given RDF triple is valid or not, we propose to find evidence for a given triple in texts. We propose to generate RDF triples from texts using an NLP pipeline, then match these to RDF triple whose validity we aim to assess. If the information is consistent between the input and extracted relations, we conclude that the RDF triple is valid according to the textual data. Moreover, the proposed method can also be employed in order to find out the missing information related to the entities that are part of the structured data set. For instance, DBpedia contains an RDF triple (dbr:Istanbul dbo:populationMetro 14,657,434). However, we have a document that is published recently that has a statement 'The most populated province was İstanbul with 15 million 29 thousand 231 inhabitants, constituting 18.6% of Turkey's population'[7] If we can extract the RDF triple (dbr:Istanbul dbo:populationMetro 15,029,231) from this text and compare it to the triple present in DBpedia, we can assess that as of 31 December 2017, the population size of Istanbul was 15,029,231 and that the old value is not valid anymore.

3.3 Validating extractions

In the 57 books that comprise the travel writings on Italy corpus, 2,226 location entities are annotated, but some locations are mentioned more than once, so we identified 903 unique location strings.

We tried to automatically disambiguate each location name using GeoNames and DBpedia knowledge bases based on string matching and DBpedia spotlight [6], respectively. Figure 2 displays the number of location entities, the number of entities linked using GeoNames and the number of entities linked using DBpedia. As the graph shows, we only find links for fewer than half the entities in either resource, with GeoNames having a slightly better coverage. This indicates gaps in the linked data resources preventing us from using the linked data resource to validate information from texts, or to further enrich them. It should be noted

[7] `http://www.turkstat.gov.tr/PreHaberBultenleri.do?id=27587`. Last retrieved 8 January 2019.

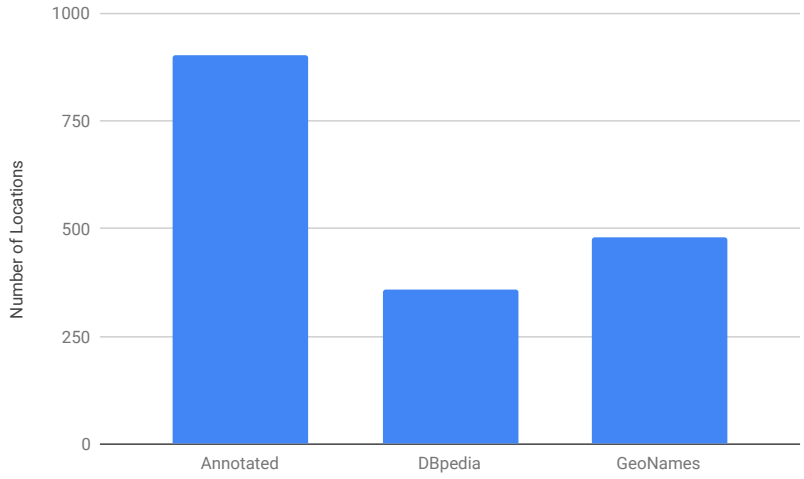


Figure 2 Number of entities and entities linked from GeoNames and DBpedia.

here that we only look at recall here, and precision is not evaluated formally so the actual number of correctly disambiguated entities is very likely lower.

An example of a recall issue is a mention of the 'chapel of San Giuliano', between 'Val di Genova' and 'Val di Borzago'[8] Many towns have chapels dedicated to Saint Julian, but this is a particular church located in the hills north of Trento. On current-day maps, this is called Rifugio San Giuliano, and neither the chapel, nor Val di Genova or Val di Borzago occur in Geonames or DBpedia. Deep NLP could help create linked data that encodes this information, although to georeference the exact locations, detailed maps, gazetteers and/or GIS sources would still be needed.

A big issue related to precision is that some location names are not unique; in the corpus, we find locations such as 'Piazza', which is used to denote the town square and can only be disambiguated in the context of knowing which town the author is talking about.

Location names are also often reused. 'Poggio', for example, as it is mentioned in 'Italian Days and Ways'[9] probably refers to Poggio San Remo because nearby in the text Taggia and San Remo are mentioned. However, in general Poggio can refer to many different places scattered around the country.[10]

In order to distinguish between different locations with the same name, entity disambiguation methods need to expand the context that they take into account and go beyond sentence or paragraph barriers (as humans do). There are efficiency concerns here, as this can be computationally expensive, but we consider this a prerequisite for true deep language understanding.

An example of a location name that is both valid in only certain contexts and ambiguous as to what it exactly refers to, is 'Monte S. Giuliano'. In the travel writings corpus, this location is described in 'Diversions of Sicily'[11] as 'This mountain, formerly world-renowned

[8] 'Italian Alps Sketches in the Mountains of Ticino, Lombardy, the Trentino, and Venetia' by Douglas William Freshfield `http://www.gutenberg.org/ebooks/45972`. Last retrieved 10 January 2019

[9] By A. Hollingsworth Wharton source: `https://www.gutenberg.org/ebooks/44418` Last retrieved 10 January 2019

[10] `https://en.wikipedia.org/wiki/Poggio` Last retrieved 10 January 2019

[11] By H. Festing Jones source: `https://www.gutenberg.org/ebooks/24652` Last retrieved 10 January 2019

as Mount Eryx, and still often called Monte Erice, is now Monte S. Giuliano and gives its name both to the town on the top and to the commune of which that town is the chief place.' According to Wikipedia,[12] the town was named back to Erice in 1934, but as 'Diversions of Sicily' was first published in 1909 and republished in 1920, the reversion back to the old name was not in there. The history of name changes is not (yet) encoded in DBpedia, GeoNames, or Pelagios[13] although it is present in the the Wikipedia page listing renamed places in Italy.[14] Analysis of this page or deep text analysis of the Erice Wikipedia page and its mention in the travel writings corpus could provide this.

4 Discussion and Conclusion

Textual documents are rich sources of information which due to their unstructured nature cannot easily be validated or updated automatically. Alternatively, linked data may contain invalid instances which can be checked with information coming from textual sources. We posit that a combination of natural language processing and linked data provides interesting opportunities for quality evaluation of both types of data.

In this paper, we proposed definitions for validity of textual data and Linked Data. We illustrated different aspects of validity through an analysis of a corpus of travel writings from the 19th and 20th centuries.

In our work, we focused on an analysis of validity issues of location names, which, whilst most locations will stay inhabited for a while, names of towns change. We suggested a combination of NLP and linked data can be utilised to check the validity of information as well as difficulties for these approaches. Whilst combining NLP and linked data is not new, our use case illustrates that this topic deserves more attention. In future work, aspects of validity for different types of information can be investigated. We will connect our analyses to research on trust and provenance on the semantic web, to assess and model trust and reliability.

Furthermore, we plan to extend our experiments by enriching the dataset with entity links such that we can assess the precision and work towards automating data validation. As our initial linking experiment showed that both DBpedia and GeoNames have insufficient coverage for historical location names, we will consider more knowledge bases to compare with and include other domains. We will investigate which properties and historical information about the extracted locations are useful to further automate the validation process.

References

1 Davide Ceolin, Valentina Maccatrozzo, Lora Aroyo, and T De-Nies. Linking Trust to Data Quality. In *4th International Workshop on Methods for Establishing Trust of (Open) Data*, 2015.

2 Davide Ceolin, Willem Robert van Hage, Wan Fokkink, and Guus Schreiber. Estimating Uncertainty of Categorical Web Data. In *Proceedings of the 7th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2011), Bonn, Germany, October 23, 2011*, pages 15–26, 2011. URL: `http://ceur-ws.org/Vol-778/paper2.pdf`.

3 Claire Grover, Richard Tobin, Kate Byrne, Matthew Woollard, James Reid, Stuart Dunn, and Julian Ball. Use of the Edinburgh geoparser for georeferencing digitized historical collections. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 368(1925):3875–3889, 2010.

[12] `https://en.wikipedia.org/wiki/Erice` Last retrieved 8 January 2019
[13] `http://commons.pelagios.org/` Last retrieved 10 January 2019
[14] `https://en.wikipedia.org/wiki/List_of_renamed_places_in_Italy` Last retrieved: 8 January 2019

4 Filip Ilievski, Piek Vossen, and Marieke van Erp. Hunger for Contextual Knowledge and a Road Map to Intelligent Entity Linking. In *International Conference on Language, Data and Knowledge*, pages 143–149. Springer, 2017.

5 Andrew McCallum. Information extraction: distilling structured data from unstructured text. *ACM Queue*, 3(9):48–57, 2005. doi:10.1145/1105664.1105679.

6 Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM, 2011.

7 Delip Rao, Paul McNamee, and Mark Dredze. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, multilingual information extraction and summarization*, pages 93–115. Springer, 2013.

8 GBIF Secretariat. GBIF Backbone Taxonomy. *Global Biodiversity Information Facility*, 2013. URL: http://www.gbif.org/species/2879175.

9 Rachele Sprugnoli. "Two days we have passed with the ancients...": a Digital Resource of Historical Travel Writings on Italy. *SocArXiv*, 2018.

10 Marieke van Erp, Robert Hensel, Davide Ceolin, and Marian van der Meij. Georeferencing Animal Specimen Datasets. *Trans. GIS*, 19(4):563–581, 2015. doi:10.1111/tgis.12110.

11 Maria Vasardani, Stephan Winter, and Kai-Florian Richter. Locating place names from place descriptions. *International Journal of Geographical Information Science*, 27(12):2509–2532, 2013. doi:10.1080/13658816.2013.785550.

12 Iris Xie and Krystyna Matusiak. *Discover digital libraries: Theory and practice*. Elsevier, 2016.

Name Variants for Improving Entity Discovery and Linking

Albert Weichselbraun
Swiss Institute for Information Science, University of Applied Sciences Chur
Pulvermühlestrasse 57, 7000 Chur, Switzerland
albert.weichselbraun@htwchur.ch

Philipp Kuntschik
Swiss Institute for Information Science, University of Applied Sciences Chur
Pulvermühlestrasse 57, 7000 Chur, Switzerland
philipp.kuntschik@htwchur.ch

Adrian M. P. Braşoveanu[1]
MODUL Technology GmbH, Am Kahlenberg 1, 1190 Vienna, Austria
adrian.brasoveanu@modul.ac.at

Abstract

Identifying all names that refer to a particular set of named entities is a challenging task, as quite often we need to consider many features that include a lot of variation like abbreviations, aliases, hypocorism, multilingualism or partial matches. Each entity type can also have specific rules for name variances: people names can include titles, country and branch names are sometimes removed from organization names, while locations are often plagued by the issue of nested entities. The lack of a clear strategy for collecting, processing and computing name variants significantly lowers the recall of tasks such as Named Entity Linking and Knowledge Base Population since name variances are frequently used in all kind of textual content.

This paper proposes several strategies to address these issues. Recall can be improved by combining knowledge repositories and by computing additional variances based on algorithmic approaches. Heuristics and machine learning methods then analyze the generated name variances and mark ambiguous names to increase precision. An extensive evaluation demonstrates the effects of integrating these methods into a new Named Entity Linking framework and confirms that systematically considering name variances yields significant performance improvements.

2012 ACM Subject Classification Information systems → Incomplete data; Information systems → Inconsistent data; Information systems → Extraction, transformation and loading; Information systems → Entity resolution

Keywords and phrases Named Entity Linking, Name Variance, Machine Learning, Linked Data

Digital Object Identifier 10.4230/OASIcs.LDK.2019.14

Funding The research presented in this paper has been partially conducted as part of the DISCOVER (`https://www.htwchur.ch/discover`) and the MedMon (`https://www.htwchur.ch/medmon`) projects, funded by Innosuisse. The work presented here has also been partially supported through the InVID project (`https://www.invid-project.eu/`) funded through the European Union's Horizon 2020 Research and Innovation Programme under GA No 687786 and through the FFG project EPOCH.

[1] Corresponding author

© Albert Weichselbraun, Philipp Kuntschik, and Adrian M. P. Braşoveanu;
licensed under Creative Commons License CC-BY

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 14; pp. 14:1–14:15

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

State of the art Named Entity Linking (NEL) systems [14] link mentions of named entities in textual content such as newspaper articles and tweets to the corresponding entities in Knowledge Bases (KB). Many of these systems excel at identifying entities in the canonical form presented in a Knowledge Base and some also accept variations (e.g., abbreviations, alternative names), but most systems do not necessarily take into account name variance, especially if it is not available in the target KB (e.g., DBpedia, Geonames). This limitation significantly lowers recall, since name variances such as *Joe Kennedy* rather than *Joseph Kennedy*, *IBM Research* or even only *IBM* for *IBM Zurich Research Laboratory*, and *SoCal/NoCal* for *Southern/Northern California* are frequently used, especially in less formal settings such as social media.

This article focuses on assessing the effect of name variance across domains, and introduces the following strategies for addressing this problem:

(i) *Obtain name variances by combining knowledge repositories.* Blending KBs requires aligning the entity identifiers used within them, triggering quality issues due to errors caused by the necessary ontology alignment tasks [14]. However, this issue can be avoided, if the links between KBs are exploited (e.g., by collecting name variants from multiple KBs, but linking them to the most used KB). The approach presented in this paper, therefore, uses graph mining to extract name variances and to integrate them into the target knowledge base.

(ii) *Algorithmic name variance generation* derives name variances from existing names by applying heuristics such as reducing the number of tokens (e.g. shorten *IBM Zurich Research Laboratory* to *IBM* or *IBM Zurich*), changing token alignment (*IBM Research* or *IBM Laboratory*), and substituting selected tokens with frequently used synonyms (e.g. *IBM Labs*).

(iii) *Name Analyzers* focus on boosting precision by marking ambiguous name variances. This paper discusses two name analyzer implementations: a) a heuristics entropy-based algorithm where tokens known to belong to certain entity types (e.g., prefixes or suffixes for organizations and locations, title for people, etc.) contribute higher entropy scores which are used for identifying ambiguous names; b) a machine learning implementation that uses support vector machines (SVM) and features that are inspired by the heuristic algorithm.

The first two approaches are targeted at increasing recall, whereas the third one improves precision. The reference implementation of the algorithms discussed in this paper draws upon Recognyze Lite, a graph-based NEL framework.

The rest of this paper is organized as follows: Section 2 describes the state of the art in graph disambiguation and the computation of name variance; Section 3 formalizes the generation and enrichment of named entity graphs for graph disambiguation and presents the architecture used to implement the suggested name variance strategies. Section 4 presents a comprehensive evaluation of the impact of name variance on NEL and discusses these results. The paper concludes with Section 5 which provides an overview of the presented and future work.

2 Related Work

The state-of-the-art and open issues in NEL are described in the overview of the TAC-KBP tasks each year [14]. Depending on the task and features that are used (e.g., strong or weak typing and/or linking, classification or clustering evaluation, etc.), NEL tasks can be defined

and evaluated in multiple ways as explained in [29], [10] or [14]. The most general situation is called NERLC (Named Entity Recognition Linking and Classification) and involves detecting not just the entities (NER), but also the links (NEL) and associated types (NEC) [14].

Knowledge Graph (KG) disambiguation is currently considered among the most effective approaches towards NEL. Several graph disambiguation NEL tools have been listed among the top performers in NLP competitions (e.g., TAC-KBP [14], OKE [19]): AIDA [11], HITS [9], Babelfy [17], AGDISTIS [29] or the multilingual version of AGDISTIS called MAG [18]. Competing approaches include statistical disambiguation (e.g., ADEL [21] or DBpedia Spotlight [4]) and neural models (e.g., Ensemble Nerd [3] for NEL).

Almost all the NEL systems have to provide at least a basic algorithm (or alternatively a set of features) for addressing the name variance problem. Some of the recently applied methods include: query expansion [8], mention-entity similarity based on keyphrases or syntax and entity-entity coherence (Milne-Witten) in AIDA [11], maximum entropy (ME) [22], synset expansion in Sematch [32], string matching via Levenshtein distances [13], Knowledge Base Embeddings [28], and ensemble neural networks [3]. Several systems that use hybrid approaches have also been developed. The HITS system [9] uses a heuristic that includes a rule-based approach for abbreviations, considers Wikipedia redirects for most common aliases, and calls to Wikipedia search functions for less common name variants. The LIEL system [26] uses language independent features like mention-entity pair features (text-based, KB link properties, Wikipedia page titles, etc.) and entity-entity pair features (overlap, title co-occurence, etc). All of these approaches struggle with missing abbreviations, names that originate in other languages, partial matches, etc. Maximum entropy [22], has been applied in Named Entity Recognition (NER) setups, therefore improvements on top of it might be needed for NEL. Popularity prior [11] is not a good metric for new entities. Synset expansion [32] can in theory help match almost all the name variance cases provided they are covered by existing KBs which rarely happens in practice. Knowledge Base embeddings [28] are dependent upon KB data quality.

Mining for name variants by combining modern KBs helps improving the coverage of entities and their name variants, but a single KB rarely provides all the information we need. DBpedia [16] does not contain special fields for name variants, but they can be collected from different fields (e.g., *dbp:wikiPageDisambiguates*, *dbo:wikiPageRedirects*, *dbp:acronym*, etc). Wikidata [6] has less factual triples for each entity than DBpedia since it has been curated manually, but it provides more triples and many name variants for each entity (through the "also known as" field). Wikidata is ideal for identifying named entities, whereas DBpedia excels at obtaining additional information about a particular entity. JRC-Names [5] is a multilingual KB that provides lists of entities and their name variants. It focuses mostly on spelling variations and covers persons and organizations, but currently does not contain any triples for locations. Geographical KBs (e.g., LinkedGeoData [27]) can also be considered good sources of name variants, provided the users are only interested in locations and are willing to combine the names from multiple fields and languages. Improving the coverage of entities and their name variances is a good technique for improving NEL, but when the entities or their name variants are missing from KB it might be best to use the entire Internet as background knowledge as described in [1].

It has to be noted that the problem of name variances is not limited to NEL or Knowledge Base Population (KBP) systems, but rather is also relevant to any field that requires matching records or names such as ontology alignment, word sense disambiguation, data linkage or slot filling tasks [12].

3 Method

This section describes Recognyze Lite, a new NEL framework that focuses on increasing recall through the use of name variance while mitigating its impact on precision. It provides a formalization of the graph generation and enrichment problem covering the tasks of adding name variances to the knowledge graph and using name analyzers for marking ambiguous name variances. Recognyze Lite provides a flexible, multi-KB NEL system that, among others, utilizes relations between entities from any given linked data source to disambiguate between correct and false candidate mentions in an unknown text.

3.1 Graph disambiguation

Similar to Usbek et al. [29] we define our approach as follows: Given a knowledge base K as a directed graph $G = (V, E)$ with vertices V and edges E. Recognyze Lite uses SPARQL queries to obtain a sub-graph $G' = (V', E')$ with the following properties:

1. $s \in V'$ and $o \in V'$, where s refers to a resource and o either indicates a resource or a literal (i.e. in this case a name used to identify a named entity)
2. for every pair $(s, o) \in E \Rightarrow \exists p : (s, p, o)$ which is denoted to as an RDF triple in G'.

The named entity disambiguation process comprises multiple sub-tasks: (i) Directed Acyclic Word Graphs (DAWGs) [25] provide fast text search within the input documents to identify candidate entities by locating mentions of their name variances. (ii) A controlled vocabulary is applied to search for potential affixes that hint on relevant entity types. (iii) These affixes are then used to remove candidate mentions that do not match the type implied by the affix. (iv) The remaining candidate entities are then linked using multiple disambiguation algorithms in sequence. In this sub-task, the relations between the candidate mentions, as well as the significance of a single mention are used to determine the best fitting network of entities. (v) Finally, Recognyze Lite transforms the accepted entities into the desired output format.

3.2 Name variance

Name variance is the problem of finding all the different names that represent a single entity within a collection of text. In theory, enriching G' with name variances improves recall, whereas adding name variance related features to the NEL extraction pipelines improves precision.

Several cases of variance have been described in the literature (e.g., [5] or [14]): (i) known aliases (*Robert Gailbraith*, a psudonym used by *J.K. Rowling*; *John Barron* for *Donald Trump*, *Mahatma Gandhi* for *Mohandas Karamchand Gandhi*) ; (ii) hypocorisms or common aliases (*Bobby* for Robert, *Liz* for Elizabeth); (iii) abbreviations (*JFK* for both *John F. Kennedy* and *John F. Kennedy International Airpot*); (iv) multilingual names (*Austria* can have different names or spelling depending on the language: in German it will be *Österreich*, in French *Autriche*, or *Ausztria* in Hungarian); (v) partial matches (names of royal figures often fall under this category; e.g., you will more often find links to *Prince Charles* instead of *Charles, Prince of Wales*). Additionally, each entity type might have its own name variance rules. People names can often include titles (*Senator*, *Judge*, etc.) or nicknames. Organization names are often abbreviated through different methods that might involve: classic abbreviations (e.g., *NBA*), cutting suffixes (e.g., *Corp* or *Inc*); removing country or branch names (*Sony Europe* might often be referred to simply as *Sony*); combining parts of words (e.g., *Nortel* instead of *Northern Telecom*). Locations have more problems with

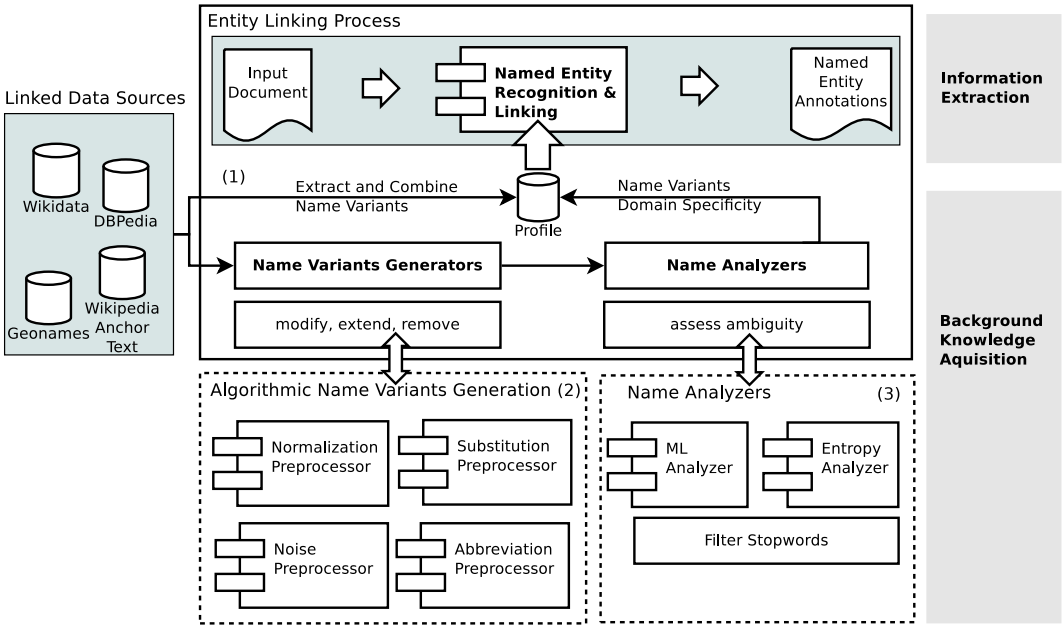


Figure 1 Name variance handling in Recognyze Lite: (1) combine name variants from multiple datasets; (2) algorithmic name variants generation; (3) name analyzers (entropy heuristic or machine learning (ML) based).

name variances than the other classes due to overlap and assimilation (e.g., people and organization names often contain location references), but can still include place qualifiers (e.g., N/E/S/W, *So* for *Southern*); regional abbreviations (e.g., *OH* for *Ohio*); embeddings or nested entities (e.g., *New York Stadium*); possessive names (e.g., *Hawaii's Waikiki*); and addresses (e.g., *221B Baker Street*).

If we take entity typing (e.g., Person – PER, organization – ORG, location – GEO, etc) into consideration, the variance problem can also include issues related to hyponyms and hypernyms [15] or even meronyms [7].

Recognyze Lite addresses the name variance problem in two ways: (i) by combining name variants from multiple datasets and (ii) by algorithmically deriving name variants from an entity's official names.

Name variances and the corresponding named entities are stored in a binary profile which is build from the knowledge base used for grounding entities. Recognyze Lite constructs knowledge graphs for NEL based on SPARQL queries that select relevant entity graphs and may comprise multiple knowledge bases (Section 3.3.1) such as DBpedia, Wikidata and GeoNames. A comprehensive preprocessing pipeline allows the analysis, manipulation and addition of name variances (Section 3.3.2), and the identification of name variances that would be harmful to the system's performance (Section 3.4).

3.3 Name variance for improving recall

3.3.1 Name variance through additional knowledge bases

The first approach for enriching the original graph draws upon further knowledge bases K_i and the corresponding graphs (V_i, E_i) to obtain tuples (s, p_j, o_k) where s is a resource in the knowledge graph G' $(s \in V')$ that is also available in knowledge base K_i $(s \in V_i)$. Adding edges $(s, o_k) \in E_i$ with relevant property types $p_j = \{p_1, ...p_n\}$ and the corresponding name variance $o_k \in V_i$ into G' enriches G' with these additional name variances.

Since such an approach might use SPARQL federation or similar technologies (e.g., RDF slicing), it is important to assess its impact on scalability before deploying it into large production systems.

3.3.2 Name variance through algorithmic name generation and assessment

The second method draws upon an algorithm $\mathcal{A}$ that splits a literal $o_k \in V'$ from the RDF triple (s, p, o_k) into tokens $t_i = \{t_1, ...t_n\}$ that are then used to generate name variances $o_k^1 ... o_k^m$ and the corresponding RDF triples $(s, p, o_k^1), ..., (s, p, o_k^m)$ to be later integrated in the knowledge graph G'.

A simple variance of $\mathcal{A}$ obtains $(n-1)$ name variances by providing substrings $o_k^1 = t_1$, $o_k^2 = t_1 t_2, ..., o_k^{n-1} = t_1 t_2 ... t_{n-1}$ of the original name. The more advanced algorithm $\mathcal{A}'$ also (i) considers synonyms by generating name variances that replace tokens t_i with synonyms $t_i^1, t_i^2, ... t_i^m$, and (ii) uses heuristics encoded in regular expressions to create name variances by modifying and reordering tokens t_i. Applying $\mathcal{A}'$ to the name "United States Department of State", for example, yields the additional name variances "U.S. Department of State" and "US Department of State". The pattern `{Department of (\w+)/\$1 Department}`, for instance, generates the name variance "Commerce Department" from the initial name "Department of Commerce". Since in many cases the abbreviations are not necessarily available in the KBs, a dedicated component is used for extracting such abbreviations directly from text such as DBpedia abstracts, if they are available.

Some preprocessing steps that are typically applied include the following: i) noise - removal of dashes, white spaces, parentheses, etc.; ii) abbreviation - for extracting abbreviations from abstracts or long texts; iii) normalization - for normalizing the entity names; iv) tickers - for detecting the company stock ticker symbols; or v) URL - removal of URLs.

3.4 Mitigating name variance's impact on precision

Name variance per se tends to improve recall at the cost of precision. We, therefore, introduce *name analyzers*, i.e. components that identify name variances which might be particularly harmful to precision.

Name analyzers aim to balance the improved recall with precision by marking ambiguous name variances, i.e. names that

1. have a high probability of clashing with common terms (e.g. *Reading*, *Turkey*, etc.) and/or
2. may clash with terms from other entity classes (e.g. *Carolina/LOC* versus *Carolina/PER*).

More formally, a name analyzer for an entity type T is considered a function $\mathcal{N}_T : o_i \rightarrow b$ that provides a mapping of name variances o_i to a binary value b indicating whether the name is considered ambiguous or not. The disambiguation process uses this information and may, for instance, require additional evidence prior to the grounding of ambiguous name variances.

Since the evaluations discussed in Section 4 are focused on news articles, we assess name variances for PER with a simple heuristic that requires at least one common English first- or surname to be present within a candidate name. For GEO we employ a simple dictionary-based list that removes names that clash with standard vocabulary.

The most challenging entity type in terms of assessing name variances are organizations for which Recognyze Lite uses an entropy-based name analyzer, as well as a machine learning approach.

The next subsections introduce these two name analyzer implementations.

3.4.1 Entropy-based name analyzer

The entropy-based name analyzer has been inspired by research from [31] and computes a heuristic entropy score that is used for assessing whether a generated name variances is considered ambiguous or not.

In information theory the entropy H specifies the minimum number of bits needed to encode sequences of random variables X produced by a probability distribution p. High entropy values, therefore, also correspond to a high diversity of values $x_i \in X$ obtained from p.

The entropy-score heuristic presented in this paper draws upon these concepts by assessing the degrees of freedom in creating valid organization names from the computed name variances (i.e. answers the question of how many *valid* organization names can be created from the available tokens). A high entropy score indicates that the name variance is very likely unambiguous, a low score, in contrast, refers to ambiguous name variances.

Tokens that are known to be used in organization names, contribute a higher entropy $H_{\text{token}}(t_j)$ (e.g. *Inc.*, *Plc.*, *AG* etc.) than tokens that are not specific to company or organization names. The heuristic also considers the number of token classes H_{classes} (i.e. abbreviation, name, legal form, etc.) used in the name variance. We compute the entropy of a name variance $\{t_i\}$ that comprises n tokens $\{t_1, t_2, ...t_n\}$ as follows:

$$H(\{t_i\}) = f_{\text{constr}}(\{t_i\}) \cdot \left[H_{\text{case}}(\{t_i\}) + H_{\text{classes}}(\{t_i\}) + \sum_{t_j \in \{t_i\}} H_{\text{token}}(t_j) \right] \quad (1)$$

The initial entropy H_{case} discounts case insensitive name variance, and the factor f_{constr} eliminates name variances that violate syntactic rules.

$$H_{\text{case}}(\{t_i\}) = \begin{cases} 0.0 & \text{if caseSens}(\{t_i\}) \\ -0.5 & else. \end{cases} \quad (2)$$

$$f_{\text{constr}}(\{t_i\}) = \begin{cases} 0.0 & \text{if } \neg\text{constr}(\{t_i\}) \\ 1.0 & else. \end{cases} \quad (3)$$

These constraints enforce that name variants (i) contain at least two characters and (ii) do not end with a connector or possessive form. This rule prevents broken names such as "Zingg &" or "Society of".

The obtained entropy measure ensures that names are unique enough to prevent ambiguities with common terminology and phrases specific to the text's language. A comprehensive corpus of ambiguous and unambiguous name variances has been used to experimentally determine suitable values for $H_{\text{case}}(\{t_i\})$, $H_{\text{classes}}(\{t_i\})$ and $H_{\text{token}}(t_j)$, to fine tune the heuristics for generating the entropy scores, and to determine the optimal threshold below which name variances should be considered ambiguous.

3.4.2 Machine learning name analyzer

We use the Java implementation of libSVM[2] to create a name analyzer that draws upon machine learning rather than heuristics for classifying name variants into ambiguous and unambiguous ones. The machine learning component considers a total of 81 features such as

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

morphological features (whether tokens are case sensitive, capitalized, all uppercase, contain letters, punctuation, etc.), syntactical features (pronouns, prepositions etc.) and semantic features (number of words mentions that refer to popular fist names, given names, trades, locations, common dictionary terms in English, French or German, etc.). Since dictionaries often also contain popular company names, a preprocessing step removes abbreviations (e.g. BBC, CNN, etc.) and the names of Forbes 2000 companies to improve their usefulness for distinguishing between common terms and potential company names.

The language-specific training corpus has been composed of (i) manually curated language-specific lists of Fortune 1000 companies, and the largest Austrian, German and Swiss companies that have been retrieved from Wikipedia, and (ii) additional 539 gold-standard entries that have been automatically derived from unit test cases used in the development of the name analyzer heuristic. A cross-validation and grid-search procedure yielded the best results for a radial basis function kernel with C=8 and γ=2^{-5}.

4 Experiments

The following section elaborates on datasets and tools used for the evaluations, the chosen evaluation settings and the evaluation results.

4.1 Datasets and Evaluation Tools

Evaluations were performed with the Orbis scorer [20], because GERBIL [30] and the neleval scorer [10] do not provide means for visually debugging results. The evaluation datasets have been selected based on the following criteria: (i) they should be available in the format, and (ii) (where possible) have been use in recent evaluation tools or challenges such as GERBIL [30] and TAC-KBP [14]. We have used two datasets included in GERBIL: N3 Reuters128 (news, multiple domains) [23] and OKE2015 (abstracts, biographies) [19].

Evaluations were performed on four state-of-the-art NEL systems which also provide REST endpoints that allow the use of sophisticated evaluation frameworks such as GERBIL and Orbis: DBpedia Spotlight [4], Babelfy [17], AIDA [11], and Recognyze Lite.

While we have tested different builds of the Knowledge Bases, the experiments described in this section used DBpedia 2015-10, Wikidata 2016-08-01 and GeoNames 2016-02-26, we preferred to use an older DBpedia version (2015-10) for the Reuters128 evaluation presented in Table 1, since the data set itself was not updated since 2014 (one year before the respective DBpedia version). This version or the one from 2014 are closer to the date when the data set was created, therefore ensuring that we are not delivering any entities that were marked as NIL (or not linked to the target KB) in the original data set, since they were not available in DBpedia at that time.

Roth et al. [24] use Wikipedia link anchor text such as *UNBRO* to expand queries for the corresponding entity (in this case *United Nations Border Relief Operation*). We apply this approach to extract additional name variances from the Wikipedia 2017-12-01 dump but only consider unambiguous link anchor text. The extracted name variances yield the Wikipedia dataset[3] used in the evaluations.

Since entity spans are to some extend dependent on a gold standard's annotation policy, we use Orbis' mention-based evaluation setting where a mention is considered correct if it (i) is found within a span that overlaps the gold standard, and (ii) refers to the same named entity as the overlapping gold standard annotation. For the gold standard sentence

[3] Available at `https://github.com/AlbertWeichselbraun/wikipedia-link-extractor`.

1. *"[Avco Corporation] has increased its profits by 10% in 2017."* where *[Avco Corporation]* refers to `dbr:Avco` both the mention *[Avco]* and *[Avco Corporation]* would be considered correct, if they refer to `dbr:Avco`.
2. The same is true for the overlapping mention *[the Netherlands]* from the sentence *"... the [Netherlands] planted a record..."* if it refers to `dbr:Netherlands`.

4.2 Evaluation Settings

The first set of evaluations demonstrates the impact of different name variance settings on the NEL performance. The *baseline* setting does not consider any name variance, operates on DBpedia only and solely uses the *rdfs:label* field for generating entity names. Setting (a) is still limited to DBpedia but considers additional DBpedia properties such as *foaf:name* and *dbp:name*. The (b1-b4) settings, draw upon multiple KBs with the intention to improve recall.

Nevertheless, the results for both the (a) and the (b1-b4) settings (Table 1) indicate that just adding additional data fields and KBs without any evaluation of name variances might even be counter productive.

Setting (c) builds upon the baseline by adding algorithmic name generation which yields considerable improvements in terms of recall at the cost of precision. The (d1-d4) settings apply algorithmic name generation to the additional KB only. The (e1-e2) configurations extend the baseline by introducing name analyzers although they are not that effective without additional name variances and, therefore, only yield significant F1 improvements for the PER type. The best performing setting (f) combines the baseline with additional properties, algorithmic name generation and Wikidata as a supplemental KB for which algorithmic name generation has been enabled as well. The heuristic name analyzer ensures a good balance between precision and recall.

Table 1 summarizes the evaluation results. We have used the R implementation of the Wilcoxon rank sum test to verify whether a particular setting yields a significant improvement at the p=0.05 significance level. Bold values indicate significant improvements, all other values are either non-significant or losses.

The second evaluation serves to illustrate that considering name variance yields competitive results. Table 2, therefore, compares Recognyze Lite's performance to three popular NEL services that offer publicly available APIs [4]. AIDA, Babelfy and Recognyze Lite use KG disambiguation techniques, while Spotlight uses statistical disambiguation. It has to be noted that each service builds its entity graph differently, therefore, not only the NEL algorithms, but also the differences between KGs can lead to variation in the results. AIDA is based on Wikipedia and, therefore, operates on a substantially different KG than the other tools. Babelfy uses the Babelnet KG and provides DBpedia links via the *owl:sameAs* property. Spotlight and Recognyze Lite both draw upon DBpedia, although Spotlight is fine-tuned for knowledge extraction tasks, whereas Recognyze Lite is optimized for NEL and various domain specific extraction tasks (e.g., Slot Filling for the recognized entities).

The Recognyze Lite baseline (Table 1) which does not consider name variance yields results that are on par with the other top systems in Table 2. Once the name variance strategies proposed in this paper are activated, the resulting system clearly outperforms all other approaches, as outlined in Table 2.

[4] Since no recommended settings for performing evaluations on Reuters128 and OKE2015 datasets have been published, we have dedicated approximately two days to experimental optimization of the evaluation settings of all evaluated third-party tools.

Table 1 Impact of name variance on the Recognyze Lite Named Entity Linking performance for the Reuter128 dataset. Bold figures indicate statistically significant improvements over the baseline.

	Setting	LOC			ORG			PER			All		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
	baseline	63	54	58	72	34	46	57	23	33	66	39	49
(a)	additional properties	63	54	58	71	33	45	57	23	33	66	38	49
(b1)	Wikidata	14	41	20	40	**41**	40	12	**38**	19	21	41	28
(b2)	Wikipedia	61	54	57	69	33	45	58	25	35	64	39	48
(b3)	GeoNames	60	54	57	71	33	45	57	23	33	64	38	48
(b4)	baseline + (b1 + b2 + b3)	14	41	21	39	**41**	40	12	**38**	19	21	41	28
(c)	algorithmic name generation	54	**72**	62	35	**53**	42	68	**49**	**57**	43	**58**	50
(d1)	name generation on Wikidata	52	54	53	71	38	50	59	26	36	61	42	50
(d2)	name generation on Wikipedia	58	52	55	68	35	46	60	29	39	63	39	48
(d3)	name generation on GeoNames	48	53	51	70	33	45	57	23	33	58	38	46
(d4)	baseline + (d1 + d2 + d3)	46	53	50	70	38	50	61	30	40	58	42	49
(e1)	name analyzer (heuristic)	64	52	57	47	**44**	46	60	**56**	**58**	54	**48**	51
(e2)	name analyzer (machine learning)	65	51	57	33	**47**	39	55	**47**	**50**	42	**48**	45
(f)	baseline + (a, c, d1, e1)	53	**70**	61	61	**52**	**57**	60	**56**	**58**	58	**58**	**58**

4.3 Discussion

Many of the settings included in Table 1 shed light on pitfalls relevant to name variance for NEL. When we designed Recognyze Lite, we proceeded incrementally, therefore expecting better results for each setting. This has not always been the case. For instance, the setting (b1) *baseline+wikidata* yields considerably worse results than the baseline profile. Initially we suspected that this effect might have been caused by data quality issues within Wikidata which is considered a relatively novel data source [6]. An analysis of the issue uncovered that the quality of Wikidata is actually high and that it yields lot of name variants per entity. This in itself is a problem as (i) gold standards usually consider a limited number of name variants for each entity, and (ii) they rarely take into account partial matches [2].

Table 2 Comparison of the system performance on the Reuters 128 and OKE2015 corpora.

Corpus	System	LOC			ORG			PER			All		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Reuters 128	AIDA	44	64	52	76	29	42	50	49	50	53	43	47
	BabelNet	29	31	30	47	16	24	21	29	24	32	22	26
	Recognyze	**53**	**70**	**61**	61	52	57	60	56	58	58	58	58
	Spotlight	41	**70**	52	64	42	51	47	22	30	50	49	49
OKE 2015	AIDA	25	37	30	69	43	53	66	41	50	50	41	45
	BabelNet	21	35	26	67	40	50	55	14	22	40	26	32
	Recognyze	**62**	**73**	**67**	70	51	59	85	57	68	73	59	65
	Spotlight	50	72	59	81	50	62	56	11	18	61	36	45

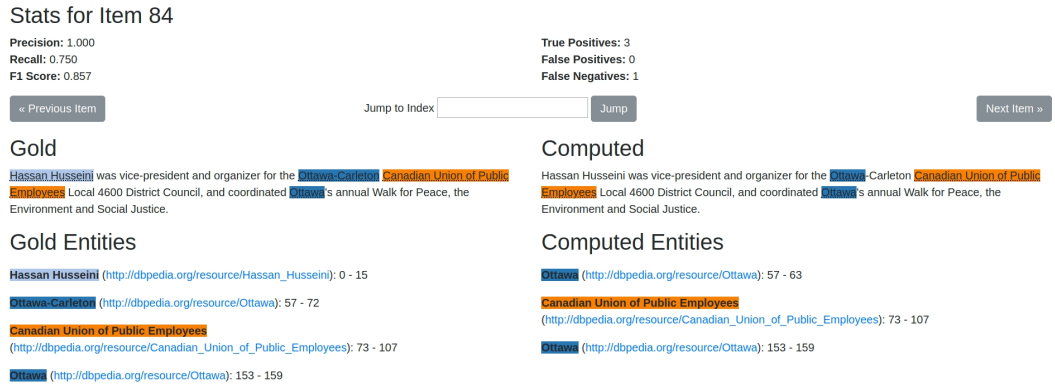


Figure 2 Debugging name variance with Orbis.

By far the most common problem was related to ambiguous name variances introduced by string splitting. Longer strings were often split into multiple entities (e.g., *Canadian Bashaw Leduc Oil and Gas Ltd* was split into *Canadian*, *Bashaw* and *Leduc*). This might not be an issue if the entity is a Person and some of the splits indicate actual roles, but if each token references a different entity (e.g. *West German Finance Minister Gerhard Stoltenberg* includes links to such ambiguous entities like `dbr:West,_Texas`, `dbr:German,_New_York` and `dbr:Minister_(Catholic_Church)`) or if there are any containment issues (e.g. *Texas Gulf Coast* is a part of *Texas*), this name variance generation strategy yields results that are similar to negative compounding. This observation triggered our research in Name Analyzer heuristics and machine learning algorithms which addresses this problem. When used in combination, both the algorithmic name generation and name analyzer components perform considerably better than the baseline+wikidata precisely because they delivered less ambiguous name variants.

DBpedia typing in itself can sometimes lead to issues, as often general terms like *stream* or *lake* might be tagged with the associated entity types, even though they are not entities. Another troubling case observed is the lack of a clear convention for embedded names (e.g., *Wells Fargo Alarm Services* embeds the name of geographical entity), geographical containment (e.g., *Texas Gulf Coast* is a part of *Texas*) or inclusion of titles in the name of entities (e.g., *chairman John Sandner* vs *John Sandner*). These problems have been especially relevant to the Recognyze Lite Wikidata and name generation evaluations (d1) presented in Table 1.

The comparison presented in Table 2 aims at providing insights into the competitiveness of the discussed name variance methods and an an assessment of whether other NEL systems could benefit from it as well. Each tool has committed a different set of errors, although the issue of ambiguous name variances due to the splitting of longer names was noticed in all tools to some degree. Most of the systems (e.g., AIDA, Babelnet) also failed to correctly identify all the name variants that belong to an entity (e.g., *Avco Financial Services*, *Avco Financial* or *Avco* can refer to the same entity). In addition, they either do not take into account abbreviations or they rarely get them correctly. In some cases, prefixes (e.g., country abbreviations – *U.S.*, *U.K.*) and suffixes (e.g., terminations like *and Co.*, *Ind.* or *GmbH*) have also created problems. Based on our analysis at least name analyzers and techniques for abbreviations would be beneficial for improving the performance of all analyzed systems.

It has to be noted that in some cases there might not be a correct way to annotate

a certain entity as illustrated in Figure 2. In this example from the OKE2015 data set, the text *Ottawa-Carleton Canadian Union of Public Employees* can be annotated as (i) *Ottawa-Carleton*, (ii) *Canadian Union of Public Employees*, (iii) *Ottawa-Carleton Canadian Union of Public Employees*, or (iv) quite possibly with an even more expanded annotation that also includes *Local 4600 District Council*. Similarly it can be argued that *Ottawa's annual Walk for Peace* should be an annotation that identifies a single recurring event. Since the results also depend a lot on the annotation guidelines of each data set, we can argue that these annotation guidelines should be openly accessible in a machine readable format (e.g., NIF, Turtle) in order to standardize evaluations and provide better comparisons between tools. Nevertheless, it needs to be noted that name variance techniques will probably not always be sufficient to address these kinds of errors, since often assigning all name variants to the correct entities is also a coreference and clustering issue.

5 Outlook and Conclusion

Considering name variances in NEL tasks significantly improves system performance. The research presented in this paper introduced three strategies for generating name variances from linked data: (i) combining knowledge repositories, (ii) algorithmic name variance generation, and (iii) name analyzers for identifying ambiguous name variances. As outlined and discussed in Section 4 these three strategies need to be deployed in concert to be effective. The use of multiple knowledge repositories or algorithmic name variance on their own does not yield significant improvements since higher recall is usually offset by lower precision or by negative effects on other entity types. Rigorous evaluations and drill-down analyses allowed understanding these issues which in turn paved the way for the development of the entropy-based name analyzer and the machine learning based name analyzer presented in this paper. These name analyzers identify and handle ambiguous name variances, substantially improving system performance. Since name variance and name analyzers can be deployed on top of existing NEL systems, the presented approach can be considered a blueprint for considerably improving the accuracy of such systems.

Future work will focus on (i) developing additional methods for identifying name variances based on deep learning, (ii) studying the effect of co-reference and clustering issues related to name variance, and (iii) better leveraging the potential of ambiguous name variances which is particularly challenging since these name variances have a high likelihood of reducing precision due to collisions with terminology used in the text that does not refer to a named entity.

References

1 Gabor Angeli, Victor Zhong, Danqi Chen, Arun Tejasvi Chaganty, Jason Bolton, Melvin Jose Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Christopher D. Manning. Bootstrapped Self Training for Knowledge Base Population. In *Proceedings of the 2015 Text Analysis Conference, TAC 2015, Gaithersburg, Maryland, USA, November 16-17, 2015, 2015*. NIST, 2015. URL: `https://tac.nist.gov/publications/2015/participant.papers/TAC2015.Stanford.proceedings.pdf`.

2 Adrian M. P. Braşoveanu, Giuseppe Rizzo, Philipp Kuntschick, Albert Weichselbraun, and Lyndon J.B. Nixon. Framing Named Entity Linking Error Types. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 266–271, Paris, France, May 2018. European

Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2018/summaries/612.html.

3 Lorenzo Canale, Pasquale Lisena, and Raphaël Troncy. A Novel Ensemble Method for Named Entity Recognition and Disambiguation Based on Neural Network. In Denny Vrandecic, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl, editors, *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*, volume 11136 of *Lecture Notes in Computer Science*, pages 91–107. Springer, 2018. doi:10.1007/978-3-030-00671-6_6.

4 Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *I-SEMANTICS 2013 - 9th International Conference on Semantic Systems, ISEM '13, Graz, Austria, September 4-6, 2013*, pages 121–124. ACM, 2013. doi:10.1145/2506182.2506198.

5 Maud Ehrmann, Guillaume Jacquet, and Ralf Steinberger. JRC-Names: Multilingual entity name variants and titles as Linked Data. *Semantic Web*, 8(2):283–295, 2017. doi:10.3233/SW-160228.

6 Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandecic. Introducing Wikidata to the Linked Data Web. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, volume 8796 of *Lecture Notes in Computer Science*, pages 50–65. Springer, 2014. doi:10.1007/978-3-319-11964-9_4.

7 Roxana Girju, Adriana Badulescu, and Dan I. Moldovan. Learning Semantic Constraints for the Automatic Discovery of Part-Whole Relations. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics, 2003. URL: http://aclweb.org/anthology/N/N03/N03-1011.pdf.

8 Swapna Gottipati and Jing Jiang. Linking Entities to a Knowledge Base with Query Expansion. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 804–813, 2011. URL: http://www.aclweb.org/anthology/D11-1074.

9 Yuhang Guo, Wanxiang Che, Ting Liu, and Sheng Li. A Graph-based Method for Entity Linking. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 1010–1018. The Association for Computer Linguistics, 2011. URL: http://aclweb.org/anthology/I/I11/I11-1113.pdf.

10 Ben Hachey, Joel Nothman, and Will Radford. Cheap and easy entity evaluation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 464–469. The Association for Computer Linguistics, 2014. URL: http://aclweb.org/anthology/P/P14/P14-2076.pdf.

11 Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust Disambiguation of Named Entities in Text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 782–792, 2011. URL: http://www.aclweb.org/anthology/D11-1072.

12 Lifu Huang, Avirup Sil, Heng Ji, and Radu Florian. Improving Slot Filling Performance with Attentive Neural Networks on Dependency Structures. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2588–2597. Association for Computational Linguistics, 2017. URL: https://aclanthology.info/papers/D17-1274/d17-1274.

13 Filip Ilievski, Giuseppe Rizzo, Marieke van Erp, Julien Plu, and Raphaël Troncy. Context-enhanced Adaptive Entity Linking. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA), 2016. URL: http://www.lrec-conf.org/proceedings/lrec2016/summaries/852.html.

14 Heng Ji and Joel Nothman. Overview of TAC-KBP2016 Tri-lingual EDL and Its Impact on End-to-End KBP. In *Eighth Text Analysis Conference (TAC)*. NIST, 2016.

15 Tomás Kliegr. Linked hypernyms: Enriching DBpedia with Targeted Hypernym Discovery. *J. Web Sem.*, 31:59–69, 2015. doi:10.1016/j.websem.2014.11.001.

16 Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015. doi:10.3233/SW-140134.

17 Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *TACL*, 2:231–244, 2014. URL: https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/291.

18 Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. MAG: A multilingual, knowledge-based agnostic and deterministic entity linking approach. *CoRR*, abs/1707.05288, 2017. arXiv:1707.05288.

19 Andrea Giovanni Nuzzolese, Anna Lisa Gentile, Valentina Presutti, Aldo Gangemi, Darío Garigliotti, and Roberto Navigli. Open Knowledge Extraction Challenge. In *Semantic Web Evaluation Challenges - Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*, volume 548 of *Communications in Computer and Information Science*, pages 3–15. Springer, 2015. doi:10.1007/978-3-319-25518-7_1.

20 Fabian Odoni, Philipp Kuntschik, Adrian M. P. Braşoveanu, and Albert Weichselbraun. On the Importance of Drill-Down Analysis for Assessing Gold Standards and Named Entity Linking Performance. In Anna Fensel, Victor de Boer, Tassilo Pellegrini, Elmar Kiesling, Bernhard Haslhofer, Laura Hollink, and Alexander Schindler, editors, *Proceedings of the 14th International Conference on Semantic Systems, SEMANTICS 2018, Vienna, Austria, September 10-13, 2018*, volume 137 of *Procedia Computer Science*, pages 33–42. Elsevier, 2018. doi:10.1016/j.procs.2018.09.004.

21 Julien Plu, Giuseppe Rizzo, and Raphaël Troncy. Enhancing Entity Linking by Combining NER Models. In *Semantic Web Challenges - Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, volume 641 of *Communications in Computer and Information Science*, pages 17–32. Springer, 2016. doi:10.1007/978-3-319-46565-4_2.

22 Livy Real and Alexandre Rademaker. HAREM and Klue: how to compare two tagsets for named entities. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, page 43, 2015.

23 Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. N^3 - A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, pages 3529–3533, 2014. URL: http://www.lrec-conf.org/proceedings/lrec2014/summaries/856.html.

24 Benjamin Roth, Tassilo Barth, Michael Wiegand, Mittul Singh, and Dietrich Klakow. Effective Slot Filling Based on Shallow Distant Supervision Methods. *arXiv:1401.1158 [cs]*, January 2014. arXiv:1401.1158.

25 Arno Scharl, Albert Weichselbraun, Max C. Göbel, Walter Rafelsberger, and Ruslan Kamolov. Scalable Knowledge Extraction and Visualization for Web Intelligence. In *49th Hawaii International Conference on System Sciences, HICSS 2016, Koloa, HI, USA, January 5-8, 2016*, pages 3749–3757. IEEE Computer Society, 2016. doi:10.1109/HICSS.2016.467.

26 Avirup Sil and Radu Florian. One for All: Towards Language Independent Named Entity Linking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. URL: `http://aclweb.org/anthology/P/P16/P16-1213.pdf`.

27 Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. LinkedGeoData: A core for a web of spatial open data. *Semantic Web*, 3(4):333–354, 2012. `doi:10.3233/SW-2011-0052`.

28 Zequn Sun, Wei Hu, and Chengkai Li. Cross-Lingual Entity Alignment via Joint Attribute-Preserving Embedding. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, volume 10587 of *Lecture Notes in Computer Science*, pages 628–644. Springer, 2017. `doi:10.1007/978-3-319-68288-4_37`.

29 Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. AGDISTIS - agnostic disambiguation of named entities using linked open data. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 1113–1114. IOS Press, 2014. `doi:10.3233/978-1-61499-419-0-1113`.

30 Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. GERBIL: General entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015*, pages 1133–1143, 2015. `doi:10.1145/2736277.2741626`.

31 Albert Weichselbraun, Philipp Kuntschik, and Adrian M. P. Braşoveanu. Mining and Leveraging Background Knowledge for Improving Named Entity Linking. In *Proceedins of the 8th International Conference on Web Intelligence, Mining and Semantics (WIMS 2018)*, Novi Sad, Serbia, 2018. ACM. `doi:10.1145/3227609.3227670`.

32 Ganggao Zhu and Carlos Angel Iglesias. Sematch: Semantic Entity Search from Knowledge Graph. In *Joint Proceedings of the 1st International Workshop on Summarizing and Presenting Entities and Ontologies and the 3rd International Workshop on Human Semantic Web Interfaces (SumPre 2015, HSWI 2015) co-located with the 12th Extended Semantic Web Conference (ESWC 2015), Portoroz, Slovenia, June 1, 2015.*, volume 1556 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015. URL: `http://ceur-ws.org/Vol-1556/paper2.pdf`.

Interlinking SciGraph and DBpedia Datasets Using Link Discovery and Named Entity Recognition Techniques

Beyza Yaman
Institute of Applied Informatics, Leipzig, Germany
yaman@infai.org

Michele Pasin
Springer Nature, London, UK
michele.pasin@springernature.com

Markus Freudenberg
Leipzig University, Leipzig, Germany
markus.freudenberg@eccenca.com

Abstract

In recent years we have seen a proliferation of Linked Open Data (LOD) compliant datasets becoming available on the web, leading to an increased number of opportunities for data consumers to build smarter applications which integrate data coming from disparate sources. However, often the integration is not easily achievable since it requires discovering and expressing associations across heterogeneous data sets. The goal of this work is to increase the discoverability and reusability of the scholarly data by integrating them to highly interlinked datasets in the LOD cloud. In order to do so we applied techniques that a) improve the identity resolution across these two sources using Link Discovery for the structured data (i.e. by annotating Springer Nature (SN) SciGraph entities with links to DBpedia entities), and b) enriching SN SciGraph unstructured text content (document abstracts) with links to DBpedia entities using Named Entity Recognition (NER). We published the results of this work using standard vocabularies and provided an interactive exploration tool which presents the discovered links w.r.t. the breadth and depth of the DBpedia classes.

2012 ACM Subject Classification Information systems → Semantic web description languages; Computing methodologies → Natural language processing; Information systems → Entity resolution

Keywords and phrases Linked Data, Named Entity Recognition, Link Discovery, Interlinking

Digital Object Identifier 10.4230/OASIcs.LDK.2019.15

Category Short Paper

1 Introduction

Scientists often search for the articles related to their research areas, however, often they fail to find the relevant publications on the search engines due to lack of semantics on document oriented search results. Thus, creating meaningful links and relations over various data sets is required to discover relevant results for the given user queries. In this paper, we describe how Linked Data technologies are applied to a publications metadata dataset from Springer Nature, such that, it is enriched with bi-directional relations to DBpedia concepts. Consequently, automatically generated semantic relations permit to construct more interesting discovery tools and contribute to the emergence of a more deeply interlinked web of data.

Springer Nature is one of the leading publishers for the educational sources and publishes large amount of articles online each year providing top-level studies to the service of the researchers but discoverability of the content is the common issue among all data sets. Thus,

© Beyza Yaman, Michele Pasin, and Markus Freudenberg;
licensed under Creative Commons License CC-BY

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 15; pp. 15:1–15:8

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

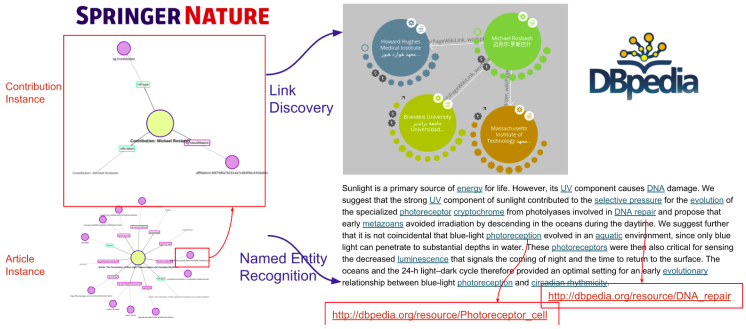


Figure 1 Two Main Approaches for interlinking SciGraph and DBpedia.

Springer Nature introduced SN SciGraph which is a Linked Open Data platform of Springer Nature Publishing and its key partners offering content from the scholarly domain. SN publishes documents and data where users can search and find the entities related to science and the scholarly domain. Platform provides around 1.5 to 2 billion triples across the research landscape dating from 1839 to 2018, e.g., funders, research projects, conferences, affiliations and publications. The model, ontology and the data sets are published under public licences providing its services to the users to explore the SciGraph data landscape in an interactive manner using SN Scigraph Explorer[1]. Moreover, data specialists can retrieve rich data descriptions for SciGraph objects by using the Linked Data API.

DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and to make this information available on the Web [1]. DBpedia data is available as Linked Data revolutionizing the way applications interact with the Web and the data sets which can serve many purposes, e.g., natural language processing, knowledge exploration, query answering. DBpedia dataset was chosen to link with SciGraph, since, it is one of the most connected and referenced data hubs on the Linked Data cloud. Not only being a data hub but also having a good categorization and type hierarchy structure convinced us that DBpedia is the most suitable data set for our use-case.

Considering these two large data sets, our main objective was to investigate application methods to enrich and improve Scigraph by employing bi-directional relations of Linked Data technologies. Thus, the contribution of this paper is three-fold: *i)* discovering links for metadata enrichment on SN articles to increase discoverability of the articles *ii)* increasing the impact of SciGraph in LOD cloud by identifying links in the existing datasets *iii)* exploring scholarly publications using DBpedia concepts. We present the applied methodologies in the following example.

Example. Fig. 1 shows above mentioned approaches illustrating on a Springer-Nature article from the Nobel Prize winner Michael Rosbash. The article's bibliographic metadata is represented as Linked Data within SN SciGraph via `sg:Article` class. This object contains information about the article's authors via `sg:Contribution` class which is used to trigger a Link Discovery algorithm and to find Michael Rosbash URI in DBpedia. This result in turns allows connecting two data sets with further links (see upper part of the Fig. 1). On the other hand, the text abstract of the article contains a wide range of keywords. They are useful to a human reader in order to have an idea about the topics mentioned in the article, however,

[1] `https://scigraph.springernature.com/explorer`

they lack formal semantics and hence cannot be interpreted effectively by machines. In order to increase the machine readability of the abstract, the text is enriched by discovering and linking these keywords to DBpedia resources via a Named Entity Recognition algorithm. (see at the bottom of Fig. 1).

In the rest of this paper, we will provide more details about the tools and methodologies adopted for these two approaches. We will discuss the obtained results and faced challenges. The remaining part of the paper proceeds as follows: Second section describes the applied methodologies in this study and the produced data sets with appropriate metadata. Third section discusses the prototype to explore publications using DBpedia concepts. The fourth section of this paper presents our conclusions, and finally, fifth section examines the possible future research directions.

2 Approach

In this section, we describe the employed techniques to interlink two data sets with relevant background and implementation details, as well as, outlining the principal results produced from the tasks.

2.1 Link Discovery

Link discovery (LD), which is considered as entity reconciliation in relational databases, is the process of automatically discovering the overlapping parts of heterogeneous data sets, and linking individual records from these data sets by exploiting their specific properties. Link Discovery is described along these lines [4]: Given two sets S (source) and T (target) of instances, a (complex) similarity measure θ over the properties of s $\in$ S and t $\in$ T, and a similarity threshold $\theta \in [0, 1]$, the goal of LD is to compute the set of pairs of instances $(s, t) \in S \times T$ such that $\gamma(s, t) \geq \theta$.

Considering this definition, we investigated some of the implemented tools specialized for Linked Data, namely, LogMap[3], KnoFuss [7], Silk[11], LIMES[6], RiMOM[10] and RuleMiner[8] to select the most convenient tool for our project. We used two frameworks to test our data set: Silk and Limes due to their advantages among other tools [5]. These advantages are high range of input types (RDF, SPARQL, CSV), various similarity measures (e.g., string, geospatial, date), ability to produce user defined links (e.g. `skos:closerMatch` while other tools only support `owl:sameAs` links), open source usage, graphical user interface, manual (rule-based description language) and semi-automatic (supervised methods) configuration possibility which allows generating links based on the similarity functions expressed in XML link specification. In the next section, we present the implementation details using this link specification configuration file.

2.1.1 Implementation Details

The interlinking process is performed by running an interlinking script with above mentioned interlinking tools between two overlapping web data sets: SciGraph *Contribution* class and DBpedia *Person* class. We produced a link specification configuration to find interlinks between instances and algorithm of the configuration which can be found in our GitHub repository. However, we have seen that Silk has a wider range of operations and transformation functions which are applied to the properties (tokenizations, lowercase etc) than Limes. Therefore, although we used Limes to test the tool and to contribute to its development, we exploited only Silk to produce links from the actual data set.

While extending the configuration file iteratively to find the best configuration, we also extended the data set with more distinctive properties. Therefore, common links between SciGraph and DBpedia data sets are increased by enriching both of them with additional properties: i) SciGraph data set is extended with properties from Orcid data set which provides a unique ID for each researcher. ii) DBpedia links are extended with unique ids from Grid data set. Thus, the links between *Affiliation* class from SciGraph and *Organization* class from DBpedia are increased by adding *Affiliation* information to the configuration. However, instead of link discovery method, these links are discovered by using link traversal methods by creating direct links between Grid organizations[2] and DBpedia Organizations discovering 30.426 links between those data sets.

2.1.2 Results

We have executed the configuration on the *Contribution* instances of the 2017 abstract articles and DBpedia *Person* instances. Since the data sizes are very large, we have limited the properties in the data sets, including only the ones used in the configuration file. Even though the framework executed for 30 days, only 11.6% of the tasks were completed, thus, we had to interrupt the execution but 47.913 links have been found in this period.

Table 1 Found links by Link Discovery approach.

Task	#SciGraph Instances	#DBpedia Instances	#Found Links
Contribution-Person	1.412.018	1.396.811	47.913 links

2.2 Named Entity Recognition

Named entity recognition (NER) is the automatic extraction process of name identification in the unstructured text. This process involves identification of proper names in texts, and classification into a set of predefined categories of interest with the possibility of connecting them to a knowledge base (DBpedia, Wikidata) to enrich the data semantically and allow to extract new connections based on created links. This structured information has the potential of deducing new inferences and arriving to the new conclusions with much more meaningful solutions, as well as, more relevant answers to the posed queries.

In the scope of this work, we first analysed the different NER tools, namely, DBpedia Spotlight[2], Stanford NER[3], AlchemyAPI [4], ANNIE [5], Open Calais [6]. Among them all, DBpedia Spotlight, which is a tool enabling automatic annotation of text documents with DBpedia URIs, is selected to conduct our experiments. DBpedia Spotlight is chosen due to its public licence, its optimal results with preliminary abstract tests and its wide range of linking possibility to the DBpedia resources chosen among more than 380 cross-domain types existing in DBpedia (e.g., people such as Obama, chemical compounds such as alkali salt or more general concepts such as humanitarian aid). The tool uses spotting, candidate selection, disambiguation and filtering respectively to discover the name entities in the text content and produces either candidate links or named entity links with requested data format, e.g. NIF, XML, JSON.

[2] https://www.grid.ac/downloads
[3] https://nlp.stanford.edu/software/CRF-NER.html
[4] https://www.ibm.com/watson/alchemy-api.html
[5] http://services.gate.ac.uk/annie/
[6] http://www.opencalais.com/

2.2.1 Implementation Details

DBpedia Spotlight provides a flexible configuration to the users according to their specific needs via DBpedia Ontology type filters, resource prominence (support) and disambiguation confidence of the spotted phrase. Type filter annotates only resources of a certain type or set of types, however, filter usage is avoided because of the interdisciplinary nature of the abstracts which might result with very restrictive outcomes. Support parameter defines the minimum number of inlinks a DBpedia resource has to have in order to be annotated where high support selects the more famous links. We configured this parameter to be low (20) to avoid the filtering of more relevant links. Moreover, we set higher confidence (0.85) for the actual data set to avoid noises after test evaluations on the abstract texts with 0.45 and 0.55 confidence.

We implemented a tool to produce the interlinks between data sets automatically for the given configuration which is openly provided to the community usage[7]. Although the tool is employed for the Springer Nature abstracts, it can be configured for any type of text to produce named entities. This tool allows analyzing the abstracts according to their topic and language, producing the links between articles and DBpedia resources. The tool has been assessed by processing the test data for analysis purposes of the abstracts with several adjustments to find optimized configuration for best results.

2.2.2 Results

In the scope of this work, 2017 article abstracts and 2017 book chapter abstracts are used from overall SN data sets: i) Articles data set is assessed by the given configuration and as a result, 187.107 abstracts are processed to identify the named entities in the content. The statistics on found entities are presented in Table 2. ii) Book chapters data set is assessed by the given configuration, thus, at the end 4880 abstracts are assessed where the statistics for the book chapters can be seen in Table 2. It is apparent from both articles and book chapters table that increase of the confidence value causes a decline on the produced number of the entities respectively. However, having more accurate links also comes with a side affect decreasing the number of the correct links as well. These data sets can be found on the GitHub repository of the project [8].

Table 2 NER Results for Articles and Book Chapters.

Data Set	Confidence	#abstracts entities	#distinct entities	# found entities	Average link per abstract	Execution time
Articles	0,85	187.107	54.077	776.424	4,14	~ 483 ms
Articles	0,55	187.107	89.138	2.841.682	8,7	~ 537 ms
Articles	0,45	187.107	274.204	3.967.124	10,26	~ 580 ms
BookChapters	0,85	4880	7.538	24.127	4,94	~ 332 ms
BookChapters	0,55	4880	12.227	45.205	9,26	~ 380 ms
BookChapters	0,45	4880	14.911	61.013	12,5	~ 434 ms

Metadata. NIF dataset is produced for each article and book chapter abstract with the prefix of the article for the named entities using NIF core ontology[9]. Moreover, provenance links are provided from the phrases to the article to reference the source of the phrase back

[7] http://hacks2019.michelepasin.org/dbpedialinks
[8] https://github.com/dbpedia/sci-graph-links
[9] http://persistence.uni-leipzig.org/nlp2rdf/

to the article as it can be seen in Listing 1. This triple shows the origin of the phrase by using `prov:hadPrimarySource` property. Such that, it allows us to traverse the phrase back to its origin source article or it would be possible to find named entities for a given article. This piece of information is included in the data set folder as well.

Listing 1 Phrase provenance link to its article.

```
<http://scigraph.springernature.com/things/articles/d8a8cee79015eecf1ff48e2edd4c27a3#
    offset_684_696> <http://www.w3.org/ns/prov#hadPrimarySource> <http://scigraph.
    springernature.com/things/articles/d8a8cee79015eecf1ff48e2edd4c27a3>
```

Parallel to the creation of the NIF dataset, also Backlinks data set is created which includes direct links from SciGraph article phrases to the DBpedia resources in the quadruple format as it is presented in Listing 2. This quadruple connects the phrase with DBpedia via `schema:mentions` property with additional information of its article, the tool it is produced by and the confidence of the tool.

Listing 2 Phrase backlink to DBpedia with confidence value.

```
<http://scigraph.springernature.com/things/articles/d8a8cee79015eecf1ff48e2edd4c27a3#
    offset_684_696> <http://schema.org/mentions> <http://dbpedia.org/resource/
    Transfection> <http://scigraph.springernature.com/things/articles/
    d8a8cee79015eecf1ff48e2edd4c27a3#nlptool=spotlight&confidence=1.0>
```

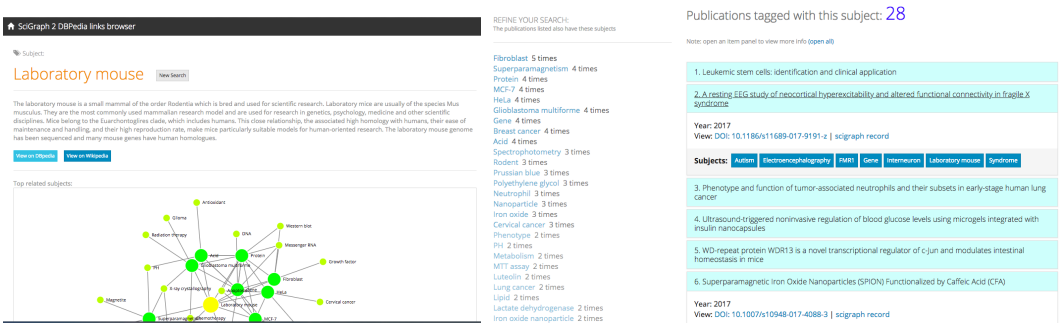
3 Application: Discovering publications using DBpedia concepts

In order to assess the relevance and usefulness of the extracted links using named entity recognition approach (see Section 2.2), a web application tool is developed that allows discovering SN publications using the DBpedia concepts they have been tagged with.

The application, which is freely available online[10], allows users to explore a subset of the data presented in this paper (87k publications tagged using 54k DBpedia concepts). An exploration journey can be initiated either by searching for a specific DBpedia concept using keywords or by listing out all of them alphabetically. Once a concept of interest has been selected, a "topic" page for that concept is presented to users, which provides a description of the concept (dynamically retrieved from DBpedia) and a list of publications tagged with that concept (Fig 2). In order to make the browsing experience more interesting and allow for a more serendipitous discovery of related content, the application presents to users other relevant concepts employing various mechanisms: first, an interactive network visualization representing the most frequent co-occurring concepts (Fig. 2.a); second, a text list of all co-occurring concepts with counts (akin to a facet search); finally, the full list of concepts related to each single publication can be displayed on-demand via a simple open/close panel widget (Fig. 2.b).

The goal of this exploration interface was to assess the relevance of DBpedia concepts via face-to-face user testing sessions involving domain experts; furthermore, it helped us shed some light on whether the kind and range of concepts available are appropriate for this kind of publication-discovery tasks. Finally, it also let us review these results with the Springer Nature ontology managers who are responsible for the (mostly manual) ongoing tagging of new content with keywords and ontology concepts. Historically, this task has been particularly time-consuming and difficult to manage, since it relies on a subject taxonomy developed in-house[11]and on the help of internal editors and domain experts.

[10] `http://hacks2019.michelepasin.org/dbpedialinks/`
[11] `https://scigraph.springernature.com/explorer/taxonomies/`



(a) Topic view of the articles with categories. **(b)** Open/close panel mechanism of the platform.

Figure 2 SciGraph Exploration Tool.

In general, despite the preliminary and informal character of these testing sessions, we still were able to gather some key findings:

- All users appreciated the breadth and depth of the concepts used to tag publications, often recognizing that it would be extremely costly to reproduce it at scale by using human annotators. Springer Nature publications simply covers too many subject areas for a manual approach to be sustainable.
- Although we used a rather high threshold for the Spotlight extraction algorithm (confidence of 0,85), we still encountered several instances of DBpedia concepts which are completely irrelevant (eg., "A roads in Zone 3 of the Great Britain numbering scheme" `http://hacks2019.michelepasin.org/dbpedialinks/entities/80611`, or "A Deeper Understanding" `http://hacks2019.michelepasin.org/dbpedialinks/entities/80649`). It's hard to speculate as to what percentage of data is wrongly annotated without a more systematic analysis. However, as a solution to this problem, it seems reasonable to assume that a mechanism to filter out extracted concepts based on the broader topic of a publication (e.g. "chemistry" or "physics") would be beneficial.
- The navigation mechanisms based on co-occurring concepts proved to be a powerful mean to explore the data set via relevant yet non-trivial pathways. In other words, they seemed to allow for a more serendipitous discovery mechanism compared to more static, taxonomy or ontology driven semantic relationships.
- Ontology managers particularly appreciated the fact that concept definitions are extracted from DBpedia automatically. Normally ontology managers spend a lot of time trying to get such definitions from subject matter experts, so they thought that using a Wikipedia definition as a starting point (or fall back) could be very valuable.
- Similarly, despite the wrongly tagged publications, ontology managers thought that often the DBpedia concepts could serve to identify under-represented areas in the corpus. Hence they could be used as candidate concepts for the official in-house subject taxonomy used at Springer Nature.

4 Conclusions and Future Work

In this paper, we have presented two approaches to increase the discoverability and reusability of the Springer Nature SciGraph scholarly data by integrating them to DBpedia, a highly interlinked data set in the LOD cloud. In order to achieve this goal, we applied techniques that a) improve the identity resolution across these two sources using Link Discovery for the

structured data and b) enrich SN SciGraph unstructured text content with links to DBpedia entities using NER. The educational publications are presented through topical navigation with specific links to DBpedia and Wikipedia to provide additional information from the open source knowledge. Overall, we strongly believe that the better connected scholar content can be highly useful for the researchers and end-users benefit from the created content.

Automated data will never be entirely accurate so mechanisms are in place for registered users to correct data when it is found to be wrong [9]. Thus, as future work, we aim at:

- evaluating the quality of the produced data sets employing crowd-sourced user feedback to produce higher quality contents.
- using these preliminary results in order to set up a more robust user evaluation study, which aims are reviewing larger sections of the concepts extracted.
- devising and testing more intelligent mechanisms to improve the accuracy of the DBpedia concepts associate to a publication: e.g. by clustering them based on general fields of studies so to be able to score them against the broader topic of a publication (which is available via journal or book level product tags).

References

1 Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

2 Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 121–124. ACM, 2013.

3 Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *International Semantic Web Conference*, pages 273–288. Springer, 2011.

4 Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. A survey of current link discovery frameworks. *Semantic Web*, 8(3):419–436, 2017.

5 Axel-Cyrille Ngonga Ngomo. On link discovery using a hybrid approach. *Journal on Data Semantics*, 1(4):203–217, 2012.

6 Axel-Cyrille Ngonga Ngomo and Sören Auer. Limes-a time-efficient approach for large-scale link discovery on the web of data. In *IJCAI*, pages 2312–2317, 2011.

7 Andriy Nikolov, Victoria Uren, and Enrico Motta. KnoFuss: A comprehensive architecture for knowledge fusion. In *Proceedings of the 4th international conference on Knowledge capture*, pages 185–186. ACM, 2007.

8 Xing Niu, Shu Rong, Haofen Wang, and Yong Yu. An effective rule miner for instance matching in a web of data. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1085–1094. ACM, 2012.

9 Yves Raimond, Michael Smethurst, Andrew McParland, and Christopher Lowis. Using the past to explain the present: interlinking current affairs with archives via the semantic web. In *International Semantic Web Conference*, pages 146–161. Springer, 2013.

10 Jie Tang, Bang-Yong Liang, Juanzi Li, and Kehong Wang. Risk minimization based ontology mapping. In *Content Computing*, pages 469–480. Springer, 2004.

11 Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Silk-a link discovery framework for the web of data. *LDOW*, 538, 2009.

lemon-tree: Representing Topical Thesauri on the Semantic Web

Sander Stolk
Leiden University, Leiden, The Netherlands
s.s.stolk@hum.leidenuniv.nl

Abstract

An increasing number of dictionaries are represented on the Web in the form of linguistic linked data using the *lemon* vocabulary. Such a representation facilitates interoperability across linguistic resources, has the potential to increase their visibility, and promotes their reuse. Lexicographic resources other than dictionaries have thus far not been the main focus of efforts surrounding *lemon* and its modules. In this paper, fundamental needs are analysed for representing topical thesauri specifically and a solution is provided for two important areas hitherto problematic: (1) levels that can be distinguished in their topical system and (2) a looser form of categorization than lexicalization. The novel *lemon-tree* model contains terminology to overcome these issues and acts as bridge between existing Web standards in order to bring topical thesauri, too, to the Semantic Web.

2012 ACM Subject Classification Information systems → Semantic web description languages; Information systems → Thesauri

Keywords and phrases lemon-tree, lemon, OntoLex, SKOS, thesaurus, topical thesaurus, onomasiological ordering, linked data

Digital Object Identifier 10.4230/OASIcs.LDK.2019.16

1 Introduction

An increasing number of dictionaries are represented on the Web in the form of linguistic linked data using the *lemon* vocabulary (e.g. [3, 12]). Such a representation facilitates interoperability across linguistic resources, has the potential to increase their visibility, and promotes their reuse [5, 13]. The core of the *lemon* vocabulary, OntoLex, has been designed to capture lexicons and to add their lexicographical knowledge to ontologies on the Web [14]. As capturing lexicographic information was not part of the primary aim of OntoLex, recent modules for *lemon* have sought to improve support for expressing such information [12, 2]. Using these modules, content of lexicographic resources can become part of the Linguistic Linked Data Cloud whilst minimizing information loss in the transition [2]. These modules, however, have explored mainly the need to represent dictionaries but not other lexicographical works such as topical thesauri. Indeed, previous research points out that additional terminology is needed for such thesauri [21]. The current paper aims to fill this gap by putting forward a novel model for this purpose: *lemon-tree*.

A topical thesaurus is a lexicographical work that organizes its lexical items according to their meaning (rather than alphabetically) by means of a topical structure [7, 9]. This overarching structure offers generic meanings to users as a starting point, which branch out to meanings increasingly specific. Once users locate the meaning which they are interested in, they are presented with the words or phrases that express that meaning. This overarching topical system in a thesaurus thus allows the user to move from meaning to lexical item [8].

The new *lemon-tree* vocabulary, described in this paper, bridges the existing standards SKOS [19] and *lemon* in order to express the content of topical thesauri on the Web. The SKOS vocabulary already allows for sharing concepts in RDF and organizing them in hierarchies. The *lemon* model and its core module OntoLex allow for sharing lexical entries, senses, and further lexicographic material. Terminology from both the SKOS and *lemon*


© Sander Stolk;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 16; pp. 16:1–16:13

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

standards, then, are valuable for sharing topical thesauri on the Web in an interoperable manner. The *lemon-tree* model therefore aims to facilitate their combined use for that purpose, adding some terminology for perceived lacunae.

2 Methodology

In order to provide insight into fundamental needs for representing topical thesauri on the Web beyond those for other lexicographic material (e.g., dictionaries), this paper will explore elements specific to the structure of topical thesauri. For each such element or structuring, the extent is discussed with which SKOS and *lemon* OntoLex offer terminology to represent these elements. For lacunae, available terminology in the new *lemon-tree* model is discussed that is fit for the purpose. Each topic is illustrated by means of an existing thesaurus that exemplifies the matter at hand. Listed in order of their appearance, these thesauri are:

- Historical Thesaurus of the Oxford English Dictionary (HTOED) [10]
- Shakespeare Thesaurus (ShT) [20]
- Scots Thesaurus (ScT) [15]
- Love, Sex, and Marriage (LSM) [4]
- Roget's Thesaurus (Roget's) [18]

Figure 1 is a legend to the images in this paper that depict the content of existing thesauri.

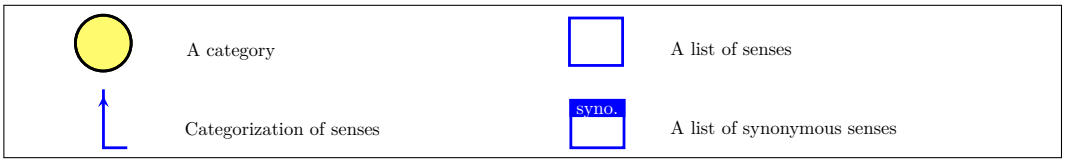


Figure 1 Legend.

Namespaces of the vocabularies relevant for this paper are provided in Listing 1. The RDF snippets in subsequent listings are specified in the Turtle RDF syntax [1]. In these snippets, samples taken from existing thesauri correspond with resources between angular brackets (that is to say, their namespace is left unspecified for the present purpose).

Listing 1 Namespaces.

```
@prefix tree: <https://w3id.org/lemon-tree#> .
@prefix ontolex: <http://www.w3.org/ns/lemon/ontolex#> .
@prefix skos: <http://www.w3.org/2004/02/skos#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

Before going to the analysis proper, the next section will first provide a short background on topical thesauri. The section that follows treats the topical system, along with the different kinds of levels distinguished in such a system. Afterwards, words and their place within the topical system are discussed, followed by the conclusion.

3 Topical thesaurus

A topical thesaurus is a lexicographic resource that organizes its items according to their meaning rather than alphabetically [7, 9]. They do this by means of a topical structure: a tree of concepts. This overarching structure offers generic meanings to users as a starting point, which branch out to meanings increasingly specific. Once users locate the meaning which they are interested in, they are presented with the words or phrases that express that meaning. This overarching topical system in a thesaurus thus allows the user to move from meaning to lexical item. Figure 2 displays the main components of such a thesaurus, using a sample of the Historical Thesaurus of the Oxford English Dictionary [10]. The senses of four nouns are shown to be categorized under "Freedom/liberty" (of which those marked with a cross no longer exist). As these four senses convey the same meaning, they are thought to be loosely synonymous.

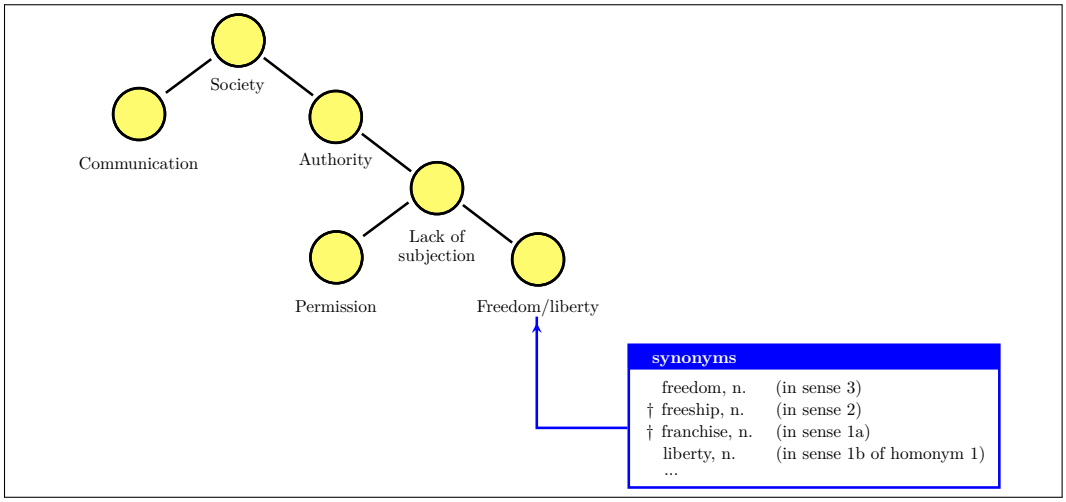


Figure 2 Thesaurus components, based on [11].

In a topical thesaurus, then, a word or phrase in a specific sense is located (or categorized) within a topical system, may be part of a set of synonyms, and is typically accompanied by additional lexicographic information such as its part of speech and usage features.

4 Topical system

The topical system of a thesaurus is its overarching structure used to organize lexical items. This structure is not unlike the taxonomies of animals and plants created by the eighteenth-century biologist Carl Linnaeus (1707-1778) and later expanded by Georges Cuvier (1769-1832) [6]. In these tree-like structures, the most generic or abstract concepts are used as roots, which branch out to concepts increasingly specific in meaning. Such topical systems can be represented with terminology from SKOS. Indeed, this standard from W3C was designed specifically for knowledge organization systems, including topical systems. Thus, the topical system as a whole would be captured as follows for the Historical Thesaurus of the Oxford English Dictionary.

Listing 2 A topical system in lemon-tree.

```
<htoed> a skos:ConceptScheme ;
    skos:prefLabel "Historical Thesaurus of the
                    Oxford English Dictionary"@en .
```

Its category "Freedom/liberty" can be captured as a SKOS **Concept**, part of the **ConceptScheme** of the topical system, and with its relation to its parent category "Lack of subjection" made explicit.

Listing 3 A category in lemon-tree.

```
<freedom-liberty> a skos:Concept ;
    skos:prefLabel "Freedom/liberty"@en ;
    skos:inScheme <htoed> ;
    skos:broader <lack-of-subjection> .
```

As we will see further on in the document, it is possible to use a specialized variant of SKOS Concept when categorizing senses. This topic will be treated in the section "Categorization and lexicalization".

4.1 Levels and depth

In a topical system, much like in any tree data structure, it is possible to distinguish multiple levels. Each level is found at a specific depth. For thesauri, however, there tend to be two forms of levels. Their topical system, after all, is meant to capture meaning and can therefore be subdivided into both levels of the tree structure and levels of meaning: tree levels and conceptual levels. SKOS and *lemon* OntoLex do not yet provide adequate terminology to capture these two levels and to distinguish them from another. The following subsections will discuss each of these levels in more detail and provides examples of how *lemon-tree* can be used to represent them on the Web.

4.1.1 Tree levels

A topical system of a thesaurus consists of categories that have been placed in a hierarchy. This hierarchical structure can be described using words for data structures known as trees. Each category in the hierarchy is a *node* in the tree, the nodes at the very top of the tree are called *roots*, and relations between nodes are known as *edges*. Each node is positioned at a certain *depth* of the tree. *Roots*, part of the first tree level, are at depth 0; nodes positioned directly below a root are at depth 1; nodes directly below these are at depth 2, and so on. Figure 3 displays such tree levels for the topical system of Roget's Thesaurus [18], perhaps the most well-known topical thesaurus in existence. Categories displayed on the same dotted line are part of the same tree level. Thus, the categories "Abstract Relations" and "Voluntary Powers" are part of the first tree level, at depth 0.

Tree levels can, of course, be calculated from the position of each node in the tree structure. Even so, some communities find it worthwhile to capture this information explicitly, too. Indeed, terminology to represent tree levels can already be found in XKOS, a vocabulary that extends SKOS [22]. In XKOS, each tree level is seen as a collection of categories, positioned at a specific tree depth. These collections are specialized SKOS **Collection**s. Although XKOS can capture tree levels in the topical system of a thesaurus, it cannot be used to capture its conceptual levels.

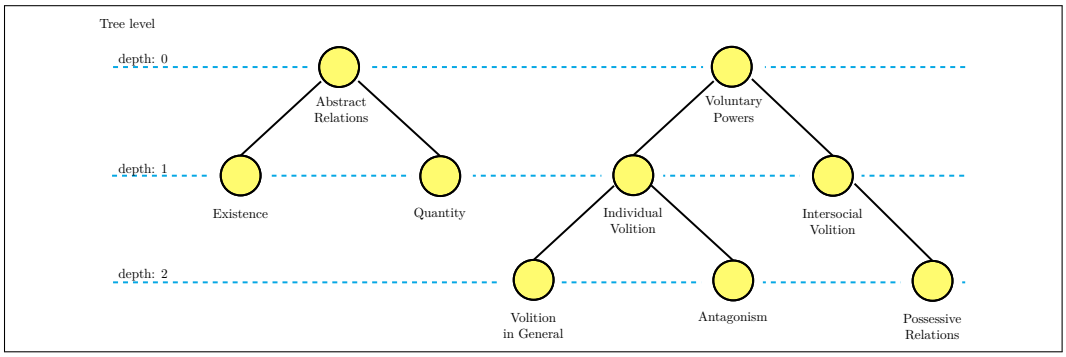


Figure 3 Tree levels (Roget's).

4.1.2 Conceptual levels

Next to tree levels, Roget's Thesaurus also contains conceptual levels. The thesaurus provides an outline of its topical system, which includes clear distinctions posited by its editor: categories in Roget's are not simply known as categories, but go by the name of *class*, *division*, or *section*. Indeed, the topical system starts out with six of these *classes*, which may branch out into *divisions* which are more specific, and ultimately into *sections*. A sample of its contents that includes these names is shown in Figure 4.

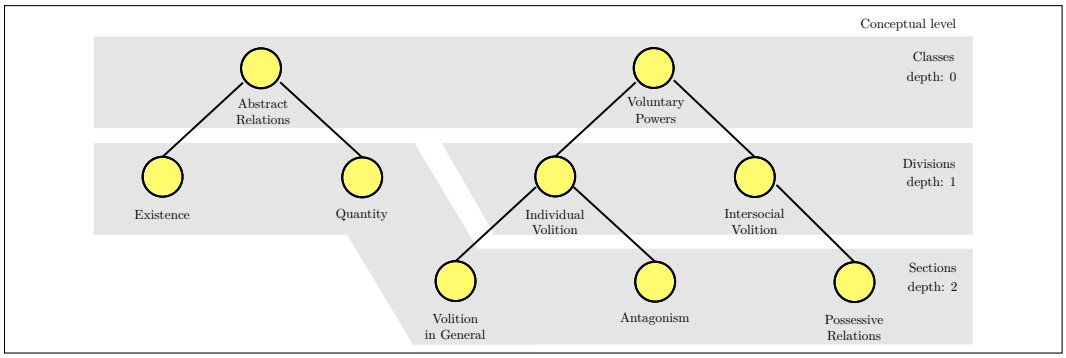


Figure 4 Conceptual levels (Roget's).

It is plain to see that the three types of category in Roget's act as a level of sorts. *Classes*, such as "Abstract Relations" and "Voluntary Powers", convey the highest level of abstraction; *sections* convey the lowest. Intuitively, categories of a higher level of abstraction branch out only to categories of a lower level of abstraction. As a consequence, we do not find categories known as *sections* in Roget's Thesaurus branching out into *classes* or *divisions*.

These levels mentioned do not necessarily map one-to-one with tree levels. In Figure 4, for example, both *divisions* and *sections* may be part of the 2nd tree level (at tree depth 1). Other thesauri, too, use similar notions to distinguish such conceptual levels [10] [4]. In the Historical Thesaurus of the Oxford English Dictionary, the first conceptual level consists of *sections*, followed by *categories* and lastly *subcategories*. Here, unlike in Roget's Thesaurus, a single category can branch out to categories from both the same conceptual level and one level beyond. A case in point is "Freedom/liberty". This is one of the so-called *categories* and branches out to a number of other *categories* (including "Independence" and "Liberation") but also to *subcategories* (including "Civil liberty" and "Moral freedom").

The *lemon-tree* model offers terminology to express these conceptuals levels. Although these levels are different from tree levels, the patterns in which thee former are captured in *lemon-tree* are analogous to how tree levels are captured in XKOS: a **ConceptualLevel** represents the level, the **conceptualDepth** property is used to indicate the depth of a level and **conceptualLevels** provides a means to list all available levels. The definitions below will be followed by snippets in which these three terms are employed.

▶ **Definition 1.** *ConceptualLevel (Class)*
A collection of concepts which are considered to be at the same conceptual depth (that is, semantically distanced from the root node). This conceptual depth may for certain thesauri coincide with the tree depth, but that is not necessarily the case for all thesauri.

SubClassOf: *skos:Collection*

▶ **Definition 2.** *conceptualDepth (DatatypeProperty)*
The depth of the conceptual level that groups a number of concepts. The conceptual depth in thesaurus taxonomies can only increase in a branch, but never decrease. The first conceptual level in a thesaurus is at depth 0; the next one at depth 1, etc.

Domain: *ConceptualLevel* ∪ *skos:Concept*
Range: *xsd:integer*

▶ **Definition 3.** *conceptualLevels (ObjectProperty)*
Provides the list of conceptual levels for a concept scheme.

Domain: *skos:ConceptScheme*
Range: *rdf:List*

Listing 4 A conceptual level in lemon-tree (Roget's).

```
<sections> a tree:ConceptualLevel ;
    skos:prefLabel "Sections"@en ;
    tree:conceptualDepth 2 ;
    skos:member <existence> ;
    skos:member <quantity> ;
    skos:member <volition-in-general> ;
    skos:member <antagonism> ;
    skos:member <possessive-relations> .

<rogets> a skos:ConceptScheme ;
    skos:prefLabel "Roget's Thesaurus"@en ;
    tree:conceptualLevels ( <classes> <divisions> <sections> ) .
```

Listing 5 A conceptual level in lemon-tree (HTOED).

```
<categories> a tree:ConceptualLevel ;
    skos:prefLabel "Categories"@en ;
    tree:conceptualDepth 1 ;
    skos:member <freedom-liberty> ;
    skos:member <lack-of-subjection> ;
    skos:member <permission> ;
    skos:member <authority> ;
    skos:member <communication> ;
    skos:member <society> ;
    skos:member <independence> ;
    skos:member <liberation> .

<htoed> a skos:ConceptScheme ;
    skos:prefLabel "Historical Thesaurus of the
                    Oxford English Dictionary"@en ;
    tree:conceptualLevels
                    ( <sections> <categories> <subcategories> ) .
```

The next section will discuss words and their place within the topical system.

5 Words and senses

A thesaurus contains lexical items that have been categorized, allowing users to go from meaning to words or phrases that express that meaning. Such words and senses can be represented using *lemon* OntoLex terminology. A word or phrase is captured as an OntoLex LexicalEntry and each of its senses as a LexicalSense. Examples thereof are presented below.

Listing 6 A lexical entry in lemon-tree.

```
<entry-freedom> a ontolex:LexicalEntry ;
    rdfs:label "freedom"@en ;
    ontolex:canonicalForm [
        a ontolex:Form ;
        ontolex:writtenRep "freedom"@en ;
    ] .
```

Listing 7 A lexical sense in lemon-tree.

```
<sense-freedom-3> a ontolex:LexicalSense ;
    ontolex:isSenseOf <entry-freedom> .
```

For further details on the notion of LexicalEntry and LexicalSense, we refer the reader to the *lemon* documentation. Advice on how to best capture other aspects of lexical items (e.g., their part of speech and other labels) is provided there, too.

5.1 Categorization

Topical thesauri do not categorize lexical items or word-forms but lexical senses: words or phrases in a particular sense. This statement may at first glance appear counter-intuitive for users of thesauri, as a number of these resources simply present head-forms of a word (or phrase) as member of their categories. In the Shakespeare Thesaurus [20], for instance, category "01.02 sky" contains the following item:

> **heaven, n.**

The head-form "heaven" in this example is similar in appearance to a headword, or lemma, found in typical dictionaries. This gives off the appearance that thesauri categorize lexical items. The following fictitious dictionary entry, however, demonstrates otherwise.

> **heaven, n.** 1) abode of one or more gods 2) the sky

It is evident that the "heaven, n." entry in the Shakespeare Thesaurus, found in the category "01.02 sky", represents the lexical item heaven in not all of its senses listed above but in only the second sense.

Werner Hüllen, who has thoroughly researched the topical tradition of thesauri, confirms that the entries in thesauri represent senses [8]. Further confirmation that topical thesauri categorize lexical senses can be found in the online edition of the Historical Thesaurus of the Oxford English Dictionary [11]. This edition takes advantage of both the topical structure of the thesaurus and the full dictionary entries of the Oxford English Dictionary. This rich set-up allows for a closer investigation of the relation between a thesaurus and entries in a dictionary. Dictionary entries in the Oxford English Dictionary have a number of senses. Each sense listed contains a reference to a thesaurus category. Conversely, the thesaurus categories in this edition list the senses they contain and provide hyperlinks not simply to dictionary entries but to specific senses *within* these entries. As such, it is evident that this thesaurus indeed categorizes senses of lexical entries, and not lexical entries as a whole. In the next section, we will provide more detail on categorization and how to capture it using *lemon-tree*.

5.2 Categorization and lexicalization

There are two forms of categorization to be found in thesauri. In the Historical Thesaurus of the Oxford English Dictionary, words in a particular sense directly express their concept. These words are said to *lexicalize* that concept. In Figure 2, "freedom" and "liberty" can directly be used if one wants to express "Liberty/freedom". This used to be the case for "freeship" and "franchise", too, in the history of the English language. (As this is no longer the case, these word senses are marked with a cross in front of them.)

Such lexicalization is not present in every thesaurus, however. In fact, it is more often the case than not in thesauri that it is absent. The sample in Figure 5 has been taken from the Scots Thesaurus [15] and illustrates this lack of lexicalization. Here, the sense 'to disperse scantily' of "blander" can hardly be said to directly express "Sowing". This is likewise the case for the sense 'a basket or container' of "happer". These senses may have a relation to the concept of "Sowing" but they do not lexicalize that concept. Their meaning causes them to be listed as part of the concept instead, that they are senses *in* that concept, as it were. Note that senses that lexicalize a concept are by definition senses also found *in* that concept. In other words, lexicalization is a special form of categorization.

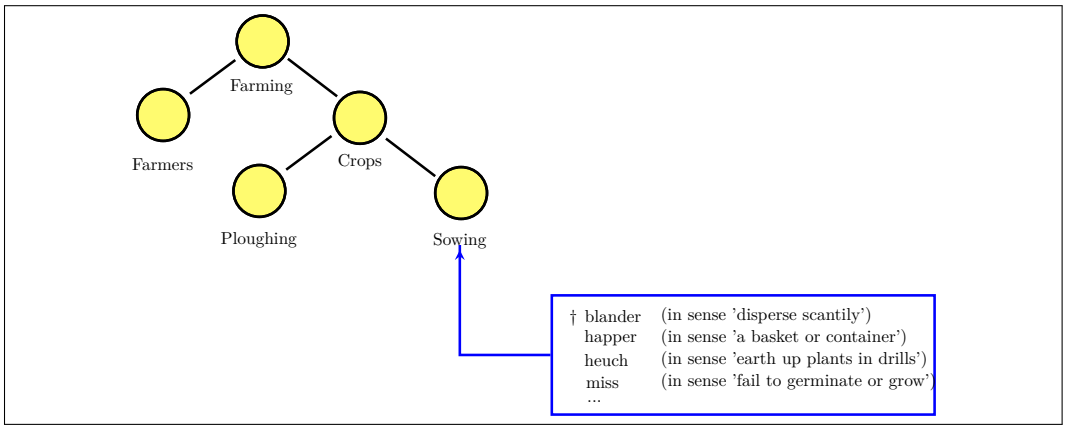


Figure 5 Sample from The Scots Thesaurus.

For asserting that senses are lexicalizations of a concept, OntoLex offers the property **isLexicalizedSenseOf**. For categorization, however, current vocabularies do not offer terminology expressive enough to capture the distinction with lexicalization and the connection between these two relations [21]. A case in point is the OntoLex property **reference**, which might appear suitable at first glance. Indeed, the property allows referring to a concept from a lexical sense. There are two problems with its use in the context of topical thesauri, however. Firstly, there is no mention in OntoLex of any direct relation between **isLexicalizedSenseOf** and **reference**, which leaves the important connection between lexicalization and categorization unexpressed and hinders inferring further knowledge from topical systems of thesauri. Secondly, the property **reference** is considered a functional one. As such, a sense may reference a single concept only using this property. However, when a sense in a thesaurus is categorized as part of a given concept, that sense is automatically also categorized as part of any parent concepts (i.e., the sense of "blander" in The Scots Thesaurus is categorized not just with "Sowing" but also with "Crops", and "Farming"). A functional characteristic therefore does not fit in this context. Instead, *lemon-tree* offers the property **isSenseInConcept** to capture these nuances needed for topical thesauri.

▶ **Definition 4.** *isSenseInConcept (ObjectProperty)*
This property relates a lexical sense to a concept that captures its meaning to some extent (that is, partially or even fully).

SubPropertyOf: *dcterms:subject*
Domain: *ontolex:LexicalSense*
Range: *skos:Concept*

The relation between **isSenseInConcept** and terminology from Ontolex has been added to the Lemon-tree model. As a result, the Ontolex property **isLexicalizedSenseOf** is asserted to be a sub property of **isSenseInConcept**. Moreover, the property **evokes** has an additional property chain of Ontolex **sense** followed by **isSenseInConcept**.

▶ **Definition 5.** *ontolex:isLexicalizedSenseOf (ObjectProperty)*

SubPropertyOf: *isSenseInConcept*
Domain: *ontolex:LexicalSense*
Range: *ontolex:LexicalConcept*

▶ **Definition 6.** *ontolex:evokes (ObjectProperty)*

Domain: *ontolex:LexicalEntry*
Range: *ontolex:LexicalConcept*
PropertyChain: *ontolex:sense o isSenseInConcept*

The examples below show how both categorization and lexicalization can be captured by employing the properties **isSenseInConcept** and **isLexicalizedSenseOf**. Notice that the property to express lexicalization is used in the example of the Historical Thesaurus of the Oxford English Dictionary. There, the use of this property automatically indicates that the category is not only a SKOS **Concept**, but a concept that is expressed or lexicalized. Such a concept is called a **LexicalConcept** according to OntoLex.

Listing 8 Categorization in lemon-tree.

```
<sense-happer-basket> a ontolex:LexicalSense ;
    tree:isSenseInConcept <sowing> .

<sowing> a skos:Concept .
```

Listing 9 Lexicalization in lemon-tree.

```
<sense-freedom-3> a ontolex:LexicalSense ;
    ontolex:isLexicalizedSenseOf <freedom-liberty> .

<freedom-liberty> a ontolex:LexicalConcept .
```

It should be noted that, whenever definitions are available for senses, it is possible to make these definitions part of the topical system. After all, the topical system allows a user to go from meaning to items that express that meaning. A sense definition is just such a meaningful item. The snippet below shows the result of this practice when applied to The Scots Thesaurus. Here, an additional concept is added to the topical system. This concept represents the sense definition of “happer” and is lexicalized by this sense.

Listing 10 Sense definitions as concepts.

```
<sense-happer-basket> a ontolex:LexicalSense ;
    ontolex:isLexicalizedSenseOf <a-basket-or-container> .

<a-basket-or-container> a ontolex:LexicalConcept ;
    skos:prefLabel "a basket or container"@en
    skos:broader <sowing> .

<sowing> a skos:Concept .
```

This approach has a caveat: synonyms are expected to lexicalize the same concept. Existing thesauri may not contain information for this additional level of grouping, requiring additional efforts in their transition to a Semantic Web form.

5.3 Synonymy

Categories in a topical system group lexical senses into sets with a similar or related meaning. In some thesauri, though certainly not all, sets exist that indicate an even stronger semantic tie: one of synonymy. A case in point is the Historical Thesaurus of the Oxford English Dictionary, in which senses placed at the same category are deemed loosely synonymous. That is to say, grouped senses in this thesaurus have a similarity in meaning and are interchangeable in specific contexts. The introduction to the thesaurus Love, Sex, and Marriage discusses synonymy found in thesauri as follows: [4]

> Grouping terms together in a thesaurus, even in a thesaurus as detailed as this, does not imply absolute synonymy. Many scholars doubt whether absolute interchangeability is actually possible.

Instead of absolute synonymy, then, it is common to find a looser form of synonymy in thesauri. This form is referred to as near-synonymy [16].

Near-synonymy is evident for lexical senses that lexicalize the same concept. After all, such senses directly express the same meaning. Thus, all the senses that lexicalize category "Freedom/liberty" of the Historical Thesaurus of the Oxford English Dictionary are known to be near-synonyms. Thus, synonymy can already be captured using terminology from *lemon* OntoLex. Using further vocabularies, it is also possible to link synonyms together via a direct relation between LexicalSenses, or to form groups of synonyms known as synsets if so desired [14].

Listing 11 Synonymy in lemon-tree.

```
<sense-freedom-3> a ontolex:LexicalSense ;
    ontolex:isLexicalizedSenseOf <freedom-liberty> .
<sense-freeship-2> a ontolex:LexicalSense ;
    ontolex:isLexicalizedSenseOf <freedom-liberty> .
<sense-franchise-1a> a ontolex:LexicalSense ;
    ontolex:isLexicalizedSenseOf <freedom-liberty> .
<sense-liberty1-1b> a ontolex:LexicalSense ;
    ontolex:isLexicalizedSenseOf <freedom-liberty> .
```

6 Conclusion

This paper set out to analyse fundamental needs for representing topical thesauri on the Web and to supply a solution for problematic areas encountered. The standardized SKOS and *lemon* vocabularies have shown to be of great value in expressing the topical system and lexical items in such a thesaurus respectively. There are, however, a few important aspects in which they fall short. The most notable two are: (1) levels that can be distinguished in a topical system and (2) a looser form of categorization than lexicalization. The novel *lemon-tree* model contains terminology to fill this gap and acts as bridge between the existing Web standards. As this paper has demonstrated, *lemon-tree* allows capturing a variety of topical thesauri –

each with its own particular characteristics. Indeed, the model has thus far been employed successfully in transitioning A Thesaurus of Old English [17] to linguistic linked data and has been found to be a good fit for the Mittelhochdeutsche Begriffsdatenbank [23]. The full data model of *lemon-tree* and its specification can be found at `https://w3id.org/lemon-tree#`.

References

1 David Beckett, Tim Berners-Lee, Eric Prud'hommeaux, and Gavin Carothers. RDF 1.1 Turtle: W3C recommendation 25 February 2014, February 2014. URL: `http://www.w3.org/TR/turtle/`.

2 Julia Bosque-Gil, Jorge Gracia, and Elena Montiel-Ponsoda. Towards a Module for Lexicography in OntoLex. In *Proc. of the LDK workshops: OntoLex, TIAD and Challenges for Wordnets at 1st Language Data and Knowledge conference (LDK 2017), Galway, Ireland*, volume 1899, pages 74–84. CEUR-WS, June 2017. URL: `http://ceur-ws.org/Vol-1899/OntoLex_2017_paper_5.pdf`.

3 Julia Bosque-Gil, Jorge Gracia, Elena Montiel-Ponsoda, and Guadalupe Aguado-de Cea. Modelling Multilingual Lexicographic Resources for the Web of Data: the K Dictionaries case. In Ilan Kernerman, Iztok Kosem, Simon Krek, and Lars Trap-Jensen, editors, *Proc. of GLOBALEX'16 workshop at LREC'15, Portoroz, Slovenia*. European Language Resources Association, May 2016. URL: `http://www.lrec-conf.org/proceedings/lrec2016/workshops/LREC2016Workshop-GLOBALEX_Proceedings-v2.pdf`.

4 Julie Coleman, editor. *Love, Sex, and Marriage: A Historical Thesaurus*. Number 118 in Costerus New Series. Rodopi, Amsterdam, 1999.

5 Thierry Declerck, Eveline Wand-Vogt, and Karlheinz Mörth. Towards a Pan European Lexicography by Means of Linked (Open) Data. In Iztok Kosem, Milo¨ Jakubíček, Jelena Kallas, and Simon Krek, editors, *Proceedings of eLex 2015*, Ljubljana/Brighton, August 2015. Trojina, Institute for Applied Slovene Studies/Lexical Computing Ltd., Trojina, Institute for Applied Slovene Studies.

6 Felipe Faria. Georges Cuvier et le premier paradigme de la paléontologie. *Revue de Paléobiologie, Genève*, 32(2):297–302, 2013. URL: `http://www.academia.edu/download/40773384/E15-Revue_Paleobiol-322-Malvesy_et_al._eds-G._Cuvier_IV.pdf#page=4`.

7 R. R. K. Hartmann. Thesauruses. In Keith Brown, editor, *Encyclopedia of Language & Linguistics (Second Edition)*, pages 668–676. Elsevier, Oxford, second edition edition, 2006. URL: `http://www.sciencedirect.com/science/article/pii/B0080448542004545`.

8 Werner Hüllen. *A history of Roget's Thesaurus: Origins, development, and design*. Oxford University Press, Oxford, 2004.

9 Christian Kay and Marc Alexander. Diachronic and synchronic thesauruses. In Philip Durkin, editor, *The Oxford handbook of lexicography*, pages 367–380. Oxford University Press, Oxford, 2016.

10 Christian Kay, Jane Roberts, Michael Samuels, and Irené Wotherspoon, editors. *Historical thesaurus of the Oxford English Dictionary: with additional material from "A thesaurus of Old English"*. Oxford University Press, Oxford, 2009.

11 Christian Kay, Jane Roberts, Michael Samuels, and Irené Wotherspoon, editors. *Historical thesaurus of the Oxford English Dictionary*. Oxford University Press, 2010. URL: `http://oed.com/thesaurus`.

12 Fahad Khan, Javier E. Díaz-Vera, and Monica Monachini. Representing Polysemy and Diachronic Lexico-Semantic Data on the Semantic Web ? In *Proceedings of the Second International Workshop on Semantic Web for Scientific Heritage co-located with 13th Extended Semantic Web Conference (ESWC 2016), Heraklion, Greece, May 30th, 2016.*, pages 37–46, 2016. URL: `http://ceur-ws.org/Vol-1595/paper4.pdf`.

13 Bettina Klimek and Martin Brümmer. Enhancing lexicography with semantic language databases. *Kernerman DICTIONARY News*, 2015.

14 Lexicon Model for Ontologies: Community report, 10 May 2016, May 2016. URL: `http://www.w3.org/2016/05/ontolex/`.

15 Iseabail Macleod, Pauline Cairns, Caroline Macafee, and Ruth Martin, editors. *The Scots thesaurus*. Aberdeen University Press, Aberdeen, 1990.

16 M. Lynne Murphy. Meaning relations in dictionaries: Hyponymy, meronymy, synonymy, antonymy, and contrast. In Philip Durkin, editor, *The Oxford handbook of lexicography*, pages 439–456. Oxford University Press, Oxford, 2016.

17 J. Roberts, C. Kay, and L. Grundy. *A Thesaurus of Old English: In Two Volumes*. Rodopi, 2000.

18 Peter Mark Roget. Thesaurus of English Words and Phrases, Classified and Arranged so as to Facilitate the Expression of Ideas and Assist in Literary Composition, 1852.

19 SKOS Simple Knowledge Organization System Reference: W3C recommendation 18 August 2009, August 2009. URL: `http://www.w3.org/TR/skos-reference/`.

20 Marvin Spevack. *A Shakespeare thesaurus*. Georg Olms AG, Hildesheim, 1990.

21 Sander Stolk. OntoLex and Onomasiological Ordering: Supporting Topical Thesauri. In *Proc. of the LDK workshops: OntoLex, TIAD and Challenges for Wordnets at 1st Language Data and Knowledge conference (LDK 2017), Galway, Ireland*, volume 1899, pages 60–67. CEUR-WS, June 2017. URL: `http://ceur-ws.org/Vol-1899/OntoLex_2017_paper_3.pdf`.

22 XKOS: An skos extension for representing statistical classifications, January 2017. URL: `http://rdf-vocabulary.ddialliance.org/xkos`.

23 Katharina Zeppezauer-Wachauer. Mittelhochdeutsche Begriffsdatenbank, 1992-2018. URL: `http://www.mhdbdb.sbg.ac.at/`.

Translation-Based Dictionary Alignment for Under-Resourced Bantu Languages

Thomas Eckart
Natural Language Processing Group, University of Leipzig, Germany
teckart@informatik.uni-leipzig.de

Sonja Bosch
Department of African Languages, University of South Africa, Pretoria, South Africa
Boschse@unisa.ac.za

Dirk Goldhahn
Natural Language Processing Group, University of Leipzig, Germany
dgoldhahn@informatik.uni-leipzig.de

Uwe Quasthoff
Natural Language Processing Group, University of Leipzig, Germany
quasthoff@informatik.uni-leipzig.de

Bettina Klimek
Institute of Computer Science, University of Leipzig, Germany
klimek@informatik.uni-leipzig.de

Abstract

Despite a large number of active speakers, most Bantu languages can be considered as under- or less-resourced languages. This includes especially the current situation of lexicographical data, which is highly unsatisfactory concerning the size, quality and consistency in format and provided information. Unfortunately, this does not only hold for the amount and quality of data for monolingual dictionaries, but also for their lack of interconnection to form a network of dictionaries. Current endeavours to promote the use of Bantu languages in primary and secondary education in countries like South Africa show the urgent need for high-quality digital dictionaries. This contribution describes a prototypical implementation for aligning Xhosa, Zimbabwean Ndebele and Kalanga language dictionaries based on their English translations using simple string matching techniques and via WordNet URIs. The RDF-based representation of the data using the Bantu Language Model (BLM) and – partial – references to the established WordNet dataset supported this process significantly.

2012 ACM Subject Classification Information systems → Resource Description Framework (RDF); Computing methodologies → Phonology / morphology; Information systems → Dictionaries

Keywords and phrases Cross-language dictionary alignment, Bantu languages, translation, linguistic linked data, under-resourced languages

Digital Object Identifier 10.4230/OASIcs.LDK.2019.17

Category Short Paper

1 Introduction

For less resourced languages, dictionary compilation is still a labour intensive task. The number of active speakers (typically between 1 and 10 million) and the number of available digital resources can be very limited: it is often difficult to collect even 100.000 sentences of raw text or get access to any enriched linguistic resources. The situation with freely available lexicographical resources is especially challenging. If available at all, the few resources are usually of questionable quality and consistency. These dictionaries are often scanned versions of dictionaries dating back a few decades. For the purpose of multilingual dictionary alignment, they often lack direct references to similar languages, but instead only


© Thomas Eckart, Sonja Bosch, Dirk Goldhahn, Uwe Quasthoff, and Bettina Klimek;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 17; pp. 17:1–17:11

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

provide inconsistent translations to European languages, like English or French. To the best knowledge of the authors no related work exists up to today that proposes a computational Linked Data-based method for aligning such multilingual fragmented and heterogeneous data for less-resourced languages. As such the presented investigation can be regarded as a promising step in building a homogeneous foundation that enables further enrichment and extension of the original data.

In this paper we will focus on examples of available dictionary sources from the Bantu language family. Many of these dictionaries have a similar, but not an identical structure: They provide word lists with varying grammatical information, translations to the target language English (or, sometimes, French), and some optional explanation in the target language. The aim of this paper is to transform this data into a unified RDF representation using the Bantu Language Model BLM. The availability of several dictionaries with different source languages, but a common target language allows the creation of aligned dictionaries using English (or French) as a pivot language. The aim of the paper is to use the Bantu Language Model to align lexical data for the three languages Ndebele [nde][1], Xhosa [xho], and Kalanga [kck] and to investigate methods which would be helpful for the generation of derived dictionaries. It will be demonstrated how the underlying graph model of the BLM enables the alignment task.

The resulting resources have the potential for a variety of use cases, like their application in all areas of language education. This is especially relevant for many Bantu languages, as their use in both primary and secondary education is currently promoted in numerous African countries, like for instance in the Republic of South Africa.

The remainder of this paper is structured as follows. Section 2 gives an overview of the Bantu language family and outlines the current situation of lexical language resources thereof. Additionally, the dictionary sources that have been used for alignment are presented. The Bantu Language Ontology as the shared modelling basis for the aligned Bantu language dictionaries is introduced in Section 3. The implementation and outcomes of the conducted RDF-based multilingual dictionary creation are then described in Section 4. Finally, a summary and prospect of future work will conclude this paper with Section 5.

2 The Bantu Language Family and Available Lexical Resources

The Bantu languages are a family of languages spoken in Sub-Saharan Africa. The total number of Bantu languages (depending on the distinction between language and dialect) is estimated at 440 to 680 distinct languages, with approximately 240 million speakers [9]. This language family represents a group of closely related languages which shows similarities in the fields of phonetics, phonology, morphology and syntax. A certain amount of common vocabulary is also involved.

The landscape of Bantu dictionary data is diverse and heterogeneous. The use of open and well-documented standards is a cornerstone for the long-term availability and reuse of existing resources, and their efficient retrieval. For example, lexicographical data for Xhosa was recently prepared and converted using a dedicated OWL ontology and is now available for all kinds of applications via standard retrieval mechanisms [1]. However, many other resources

[1] We refer to languages by their names as presented in the Ethnologue (`https://www.ethnologue.com`) and also indicate their particular ISO 639-3 codes: Xhosa [xho] is referred to as "isiXhosa", Ndebele [nde] as "isiNdebele" and Kalanga [kck] as "Kikalanga" by their respective speakers. It is important to differentiate between so-called Zimbabwe Ndebele [nde] spoken mainly in Zimbabwe, and Southern Ndebele [nbl] spoken in South Africa.

are already available in a heterogeneous digital format. One such valuable source is the Comparative Bantu OnLine Dictionary (CBOLD), which offers Bantu language dictionaries under an open licence, including data for Zimbabwean Ndebele [10] and Kalanga [7].

Two of the languages under discussion are cross-border languages. Kalanga is spoken in eastern Botswana and western Zimbabwe and has a total of 338,000 users[2]. While Kalanga is a minority language in Botswana with no official status [8, p.176], it is an officially recognised language in Zimbabwe[3]. Kalanga is classified as S16 in Guthrie's larger Shona group of languages (S10) [9, p.609]. Zimbabwean Ndebele is spoken by approximately 1.6 million people in Zimbabwe, Botswana and Zambia [4], and is also officially recognized in Zimbabwe. Xhosa, an official language in South Africa, has approximately 8.1 million speakers and is spoken predominantly in the Eastern Cape and Western Cape regions of the country. According to the new updated Guthrie classification of Bantu languages list [9, p.648], Zimbabwean Ndebele (S44) and Xhosa (S41) are classified as members of the Nguni group (S40).

These three languages all being members of the Bantu language family, in particular of the S group of languages, share many linguistic features – for instance, they are structurally agglutinating and are therefore characterised by words usually consisting of more than one morpheme. They adhere to the typical Bantu languages nominal classification system according to which nouns are categorised by prefixal morphemes. For analysis purposes, these prefixes have been sorted into classes and given numbers by scholars who have worked within the field of the Bantu language family. A total of 24 noun classes is recognized [9, p.108], but these are not all attested in any single Bantu language. Noun prefixes usually indicate number, whereby the uneven class numbers indicate singular and the corresponding even class numbers indicate plural. However, exceptions to this rule also occur, e.g. mass nouns such as "water" in so-called plural classes do not have a singular form; plurals of class 11 nouns are found in class 10, while a class such as 14 is usually not associated with number at all. Irregular pairing also occurs occasionally, e.g. classes 9/6:

Table 1 Ndebele (excerpt from Pelling's Ndebele dictionary, source: CBOLD).

Prefix	Noun stem	Lexeme	Sg./Pl.	POS	Gloss	Comments
in	simu	in-simu	in/ama	n.	(pl. ama-simu): field;	[classes 9/6]
u	suku	u-suku	ulu/izin	n.	day.	[classes 11/10]
ubu	thongo	ubu-thongo	ubu	n.	sleep.	[class 14]

Table 2 Kalanga (excerpt from Mathangwane's Kalanga dictionary, source: CBOLD).

Prefix	Noun stem	Tone	POS	Class	Gloss
	bhaisikili	LLHHH	n	9/6	bicycle
lu	nji	H	n	11/10	knitting needle; [...]; an injection needle
bu	nyambi	LH	n	14	neatness; skilfulness; cleverness

[2] https://www.ethnologue.com/language/kck
[3] Cf. https://www.constituteproject.org/constitution/Zimbabwe_2013.pdf

Table 3 Xhosa (excerpt from Louw's Xhosa data set).

Noun stem	POS	Sg. prefix	Class	Pl. prefix	Class	Gloss
khitshi	noun	i	9	ama	6	kitchen
phahla	noun	u	11	ii	10	roof
phuthuphuthu	noun	ubu	14			hastiness

It is notable that, in contrast to the other two languages under discussion, Kalanga has an additional class 21, employed to express the augmentative by means of the class 21 prefix *zhi-*, as illustrated in Table 4.

Table 4 Kalanga (excerpt from Mathangwane's Kalanga dictionary, source: CBOLD).

Prefix	Noun stem	Tone	POS	Class	Gloss	Comment
zhi	nyala	HL	n	21	thumb; big toe	(compare with: *chi-nyala*: a finger; a toe)
zhi	midza-mbila	LLLH	n	21	huge mamba snake	

In the Nguni language group, augmentation is usually indicated by means of a noun suffix which does not influence the noun class, as illustrated in the following Xhosa example:

um-thi (class 3) "tree" > um-thi-kazi (class 3) "big tree"

Like most Bantu languages, Zimbabwean Ndebele, Kalanga, and Xhosa are considered resource scarce languages, implying that linguistic resources such as large annotated corpora and machine-readable lexicons are not available. Moreover, academic and commercial interest in developing such resources is limited. In the following section, some of the available sources for lexicographical data for Bantu languages are described in more detail.

2.1 Comparative Bantu OnLine Dictionary

The Comparative Bantu OnLine Dictionary (CBOLD[4]) project started in 1994 to create a source for lexicographical data for Bantu languages. It is committed to open access principles as stated in the "Bantuists' Manifesto" [2]. Between 1994 and 2000, a large number of Bantu dictionaries were digitized by CBOLD and provided via the project Web page for external use and applications.

The amount and range of available data, and its quality vary from dictionary to dictionary. For many dictionaries, information about the respective Bantu noun classes and morphological structure is available. There is no interlinkage between lexical items of different dictionaries; an alignment is therefore not directly feasible. However, all datasets contain translations to either English or French.

Despite the completion of the project in the year 2000 with no further updates since, it is still one of the most comprehensive sources for lexicographical data of Bantu languages. The list of supported languages contains – among many others – Swahili, Zimbabwean Ndebele, Venda, and Kalanga.

The CBOLD dictionaries are provided in inconsistent data structures and schemata using a variety of file formats, including FileMaker databases, HyperCard[5], Microsoft Word documents and plain text files. Obviously, this schematic and technical heterogeneity can

[4] `http://www.cbold.ish-lyon.cnrs.fr/`
[5] A proprietary hypertext format created by Apple Inc. in the 1980s.

not be used as a basis for modern cross-dictionary alignment and inter-lingual applications. As a consequence, transformation and quality assurance measures are required to allow the active usage of this valuable lexical data source in the future.

In the following sections, two of the included CBOLD dictionaries (Kalanga and Ndebele) are described in more detail.

2.1.1 Ndebele Dictionary

The CBOLD dictionary for Zimbabwean Ndebele was compiled by James N. Pelling [10] in 1971. CBOLD provides the data as plain text file and a FileMaker database. The dictionary contains 5000 lexemes with information about the part of speech, prefix/stem structure for the nouns, translations to English and corresponding forms in the perfect passive.

For this submission, only nouns and verbs were considered. This includes 4632 of the provided lexemes (i.e. 92.6%). Table 5 shows an excerpt of the available data.

Table 5 Excerpt from Pelling's Ndebele dictionary (Source: CBOLD).

Prefix	Stem	Lexeme	Prefix	POS	Gloss	Perfect Passive
is	ayobe	is-ayobe	isi/izi	n	spider	
ama	ququ	ama-ququ	ama	n	bad smell, stench	
	cutha	-cutha		v.t.	pluck feathers	cuthwa
	finyeza	-finyeza		v.t.	shorten	finyezwa

2.1.2 Kalanga Dictionary

The CBOLD dictionary for Kalanga was created 1994 by Joyce Mathangwane [7]. CBOLD provides the data as plain text file and a FileMaker database. The dictionary contains 2960 lexemes with information about the part of speech, tone, noun classes and prefix/stem structure for the nouns. Additionally, English translations are provided.

For this submission, only nouns and verbs were considered. This includes 2796 of the provided lexemes (i.e. 94.5% of all). Table 6 shows an excerpt of the available data.

Table 6 Excerpt from Mathangwane's Kalanga dictionary (Source: CBOLD).

Prefix	Stem	Tone	POS	Class	Gloss
chi	ako	LL	n	7	corn head
m	bala	HH	n	3	colour
	anga	LL	v		freeze; congeal
	baka	HH	v		build; construct

2.2 Xhosa Dictionary

Since CBOLD dictionary data is not available for all Bantu languages, Xhosa data used for this publication was taken from a resource compiled by J.A. Louw (University of South Africa UNISA) which is available under a Creative Commons (CC) license. This Xhosa lexicographical data set consists of morphological information accompanied by English translations. It was created and made available by the authors for purposes of further developing Xhosa language resources [1]. The data were compiled with the intention of

documenting Xhosa words and expanding existing bilingual Xhosa dictionaries by means of among others botanical, animal names, grammar terms, modern forms etc., as well as lexicalisations of verbs with extensions. The publication process involved digitisation into CSV tables and several iterations of quality control in order to make the data reusable and shareable. Since this process has not yet been completed, we concentrate in this paper on two word classes for which extensive results already exist, namely nouns and verbs.

The excerpt of the lexicographical data set is a representative sample of Xhosa nouns and verbs. Nouns of all possible regular and irregular combinations of noun classes, and verbs with a variety of verbal extensions (leading to lexicalisations in meaning) are represented. Nouns are listed alphabetically according to noun stems, followed by the POS, the surface form of the singular and plural class prefixes (if applicable) as well as the number(s) of the class prefixes, and finally the English translations, like shown in Table 7.

Table 7 Excerpt of nouns from the Xhosa dictionary.

Noun stem	POS	Class pref sg	Class no.	Class pref pl	Class no.	English translation
phathi	noun	um	1	aba	2	superintendent

Verbs are listed alphabetically according to verb stem, i.e. the basic verb root followed by the inflection suffix -a, or sometimes -i, like shown in Table 8.

Table 8 Excerpt of verbs from the Xhosa dictionary.

Verb stem	POS	English translation
mi	verb	be standing
tyalisa	verb	help to plant

The lexicographic data is by no means based on corpus frequencies of nouns and verb stems as for instance the Oxford School Dictionary [3] but rather on complementation of existing, established dictionaries.

3 The Bantu Language Model

Aligning lexical content requires semantic and structural consistency between two or more language datasets. In the case of Bantu languages, as already explained, no shared structural basis for representing lexical data exists to date. The available digital resources are highly heterogeneous with regard to their size, content and format. In order to undertake any kind of alignment task these resources need to be transformed into a shared format first. While this can be done by using structured formats such as XML or entering and maintaining the lexical data in a database we decided to apply the Linked Data framework and reuse the Bantu Language Model (BLM)[6]. This model is an ontology that was introduced in Bosch et al. 2018 [1] in the RDF and OWL formats that ensure semantic and structural interoperability between all data that is described with it. An overview of the BLM is illustrated in Figure 1 which shows the underlying graph that integrates and unifies all data that is created based on the BLM. The BLM allows for the representation and interrelation of lexical, morphological and translational elements but also common grammatical meanings as well as noun class elements of Bantu languages. This is in accordance with the content

[6] The URL of the ontology is: `http://mmoon.org/bnt/schema/bantulm/`

that we found in existing tabular lexical data of various Bantu languages and with the three language datasets that were just described. More details on the underlying development and design decisions of the ontology are discussed in [1][7]. The applicability of this ontology has been proven by using it to create a Xhosa RDF dataset[8].

The choice of the BML as a suitable modelling basis that facilitates dictionary alignment is motivated by a number of aspects. First, this ontology is already specified for the peculiarities of Bantu languages, and above all, it was created together with Bantu language experts. In this way, semantic coherence between lexical elements is already ensured on the data representation level. Second, the Linked Data approach allows for the separate development of single language resources that can be later integrated and interrelated, if desired, within one unified graph due to the shared vocabulary. A third advantage is entailed in the possibility to not only interconnect various Bantu language datasets with each other but also extend the data with already existing other language resources, i.e. available English or French Wordnet RDF editions that are useful as a pivot language for identifying translations. What is more, the BLM ontology can be easily extended according to representational needs. It is not a fixed model but can be later on modified to include elements and relations that might be necessary for describing a more detailed language dataset. Finally, with regard to the practical aspect of transforming, editing, merging and analysing existing lexical Bantu resources, the compliance to the Linked Open Data framework is an additional decisive factor for the BLM, because various tools for enriching or analyzing RDF-based linguistic data already exist.

4 RDF-based Dictionary Alignment

4.1 Technical Implementation

All three dictionaries mentioned in section 2 were transformed into the RDF format by using the Bantu Language Model. The Xhosa RDF dataset could be reused directly, while for the Ndebele and Kalanga data transformation code was used, that had already generated the Xhosa RDF dataset[9]. As a result, links between English translation resources and their respective lexical WordNet resources have also been established within those two datasets. Due to missing data[10] or additional data[11], the implementation had to be adapted insignificantly. For example, temporary noun and number classes were introduced to the data set, that still have to be replaced by their correct classes during future quality assurance and enhancement procedures. Similar requirements exist for enhancing the quality of translations. For those procedures, the still ongoing work of double checking the Xhosa dataset by native speakers can be seen as a template.

The resulting RDF datasets were imported into a SPARQL endpoint[12] where they are publicly available and where future updates will also take place. All results included in the next subsection were extracted using SPARQL queries and are therefore easily reproducible.

[7] There, also the question why the OntoLex-Lemon model as widely accepted recommendation for representing lexical language data has not been used instead is answered and shall not be addressed in this publication again.

[8] The data is available here: `https://github.com/MMoOn-Project/OpenBantu/blob/master/xho/inventory/ob_xho.ttl/`

[9] The code will be available at the GitHub repository of the MMoOn project (`https://github.com/MMoOn-Project`) soon.

[10] This includes explicit noun class information for Ndebele or information about number for both Ndebele and Kalanga.

[11] Like information about tone for Kalanga.

[12] `https://rdf.corpora.uni-leipzig.de/sparql`

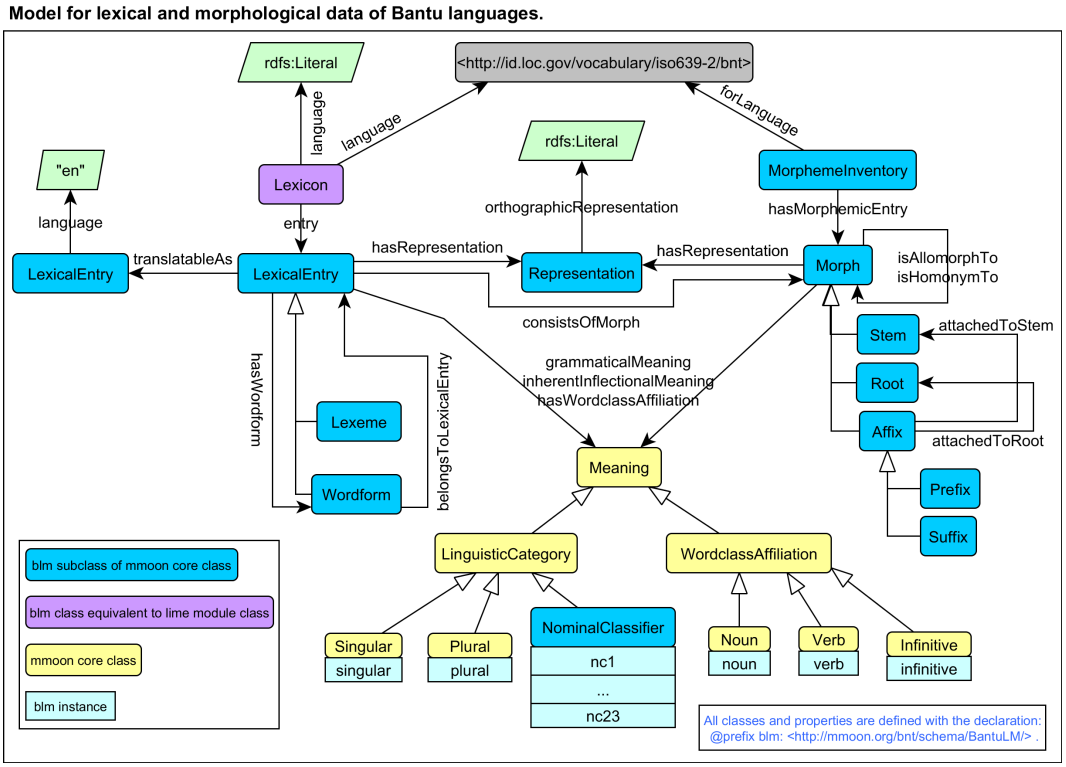


Figure 1 Ontology for the Bantu Language Model.

The actual alignments were not persisted in the endpoint, as quality assurance measures are not finished yet.

The underlying graph model of the RDF-based BLM ontology made the aggregation of the first results especially easy and is seen as a well-defined but still flexible backend for future, more user-friendly applications by the authors. First work on integrating the endpoint into an existing Web portal for lexicographical data has already shown some positive results.

4.2 Alignment Methods and Results

The identification of translations between lexical resources is already a challenging task if extensive data exists. In general, translation equivalences between two lexical entries are established if both entries share the same conceptual description, e.g. sense resources or definitions. For RDF-based datasets such an alignment between multiple dictionaries has been undertaken for the Apertium Bilingual Dictionaries [6]. While such an encompassing sense-based alignment is not feasible due to the outlined shortcomings of the source data for the three Bantu RDF language datasets under investigation, however, the demonstrated usage of pivot languages for aligning dictionaries with no direct translations was applicable for this case. Moreover, it should be noted that this contribution focuses on providing a foundation for the further enrichment of the aggregated data using a common data model. It presents work in progress and is seen by the authors as a first step towards the integration of more comparable languages. For this reason, a deeper evaluation of the results or the discussion of borderline cases was postponed to a later date.

Due to the underlying shared BLM vocabulary, semantic coherence between the lexical elements and translations between the three Bantu language RDF datasets is ensured. Loading all datasets into a single SPARQL endpoint, as done, then renders a unified data graph that

can be analysed and traversed along the nodes and edges across the three dictionaries. For the alignment only the lexeme, translation and WordNet resources could be used, since no sense definitions exist within the data. Provided with these resources we identified two methods for finding alignments. Similarly to the Apertium Bilingual Dictionaries we made use of the English translations contained in all three datasets as the pivot language interconnecting the Bantu dictionaries.

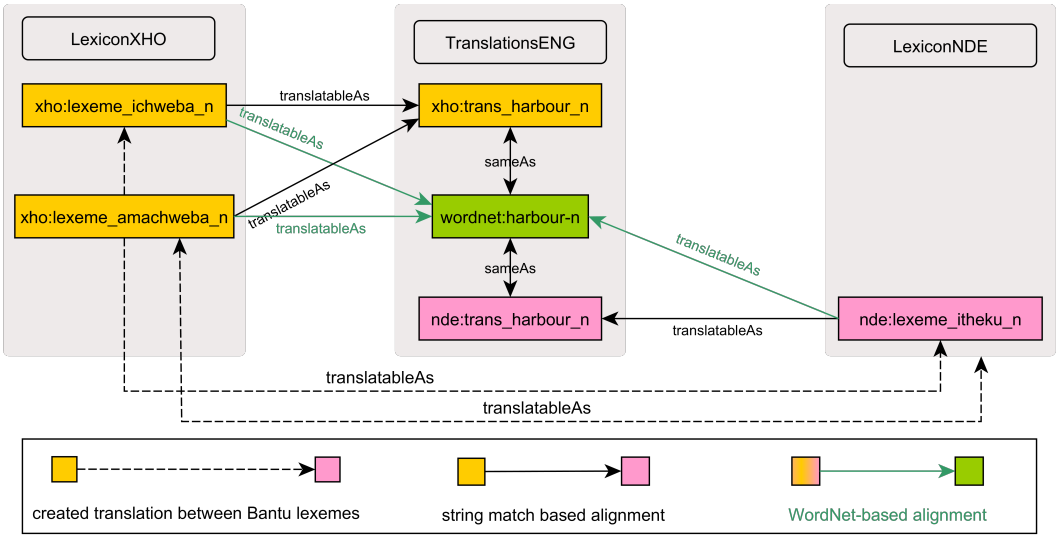


Figure 2 Example translation between lexemes in Xhosa and Ndebele in BLM RDF.

For the first method we aligned lexical entries based on the contained WordNet data [5], that is two lexical entitites are considered as translations if they point to the same WordNet resource. Since the English translation resources are interlinked with a WordNet resource via the `owl:sameAs` object property also a direct translation between a Bantu language lexeme and this WordNet resource can be inferred. The second method involves the identification of translations for which no shared WordNet resource exists. By conducting a simple string match between all translation resources across the three dictionary pairs, an alignment between lexical entries could be obtained whenever the strings of two English translation resources of different dictionaries were identical. Both methods are illustrated in Figure 2. As can be seen, the WordNet-based alignment contains the string match based alignment in that the WordNet links were also created based on string match with the English translation resources. While this seems to occur redundant we explicitly represent this method here because we regard the identification of translations by pointing to a single English dataset, which is the English WordNet in this case, as more accurate than the string match based alignment. In this special case for available Bantu language data the prospective creation of more BLM-based RDF dictionaries will result in a number of duplicate and ambiguous English translation strings without any further lexical information, e.g. `xho:trans_harbour_n` and `nde:trans_harbour_n`. Indeed, as the number of resulted translations in the three bilingual dictionary pairs in Table 9 show, there could be only one more translation for the Ndebele-Kalange and Xhosa-Kalanga dictionaries and just 67 translations for the Ndebele-Xhosa dictionary obtained via the string match based method in addition to the WordNet-based method.

Table 9 Available alignments for all dictionary pairs.

Dictionary pair	WordNet-based alignments	String-matching alignments
Ndebele, Xhosa	1541	1608
Ndebele, Kalanga	62	63
Xhosa, Kalanga	106	107

Consequently, we regard the WordNet-based method as more suitable for retrieving translations. Creating links from translations of single Bantu dictionaries to one shared and already existing dataset, such as the English RDF WordNet, facilitates the quality assessment of obtained alignments by language experts because WordNet also comes with definitions which can be used to ensure that the right translation has been found. Moreover, WordNet provides lexical entries with sense resources which could be used to arrive at more accurate sense-based translations in the future.

In addition to the bilingual translation data that was found, multilingual translations between all three dictionaries could be identified using the same methods (cf. Table 10). By that, it could be shown that analysing lexical data in the RDF format is very simple and efficient since every data point is interconnected and retrievable by traversing the graph. The quality of the established alignments with regard to their linguistic accuracy cannot be evaluated at this stage since it is future work to be done by language experts. Nevertheless, we judge the resulted numbers of obtained alignments across the bilingual dictionaries as promising. Taking into consideration that the strings of the English translation resources were the only available information usable as a comparative measure between Bantu language lexemes, the presented alignments can be considered as the closest one can get to bi- and multilingual translations for Bantu language data given the current state of the language data situation. What is more, the outcome of this translation-based dictionary alignment provides valuable additional data for the less-resourced Bantu languages that is easy to obtain and directly usable by language experts.

Table 10 Examples for aligned lexemes in all three source languages.

English	Xhosa	Kalanga	Ndebele
companion	iqabane	nkwinya	umngane
debt	isikweliti	nlandu	isikwilidi
doctor	ugqirha	nlapi	udokotela
image	umfanekiso	itshwantsho	isithombe
witch	igqwirha	nloyi	umthakathi

5 Conclusion

The presented prototypical implementation for aligning Xhosa, Zimbabwean Ndebele and Kalanga language dictionaries revealed typical problems of this task for less-resourced languages. While there is a need for aligned data, the available dictionaries are typically unsatisfactory concerning size, quality and consistency, which makes interconnecting them to form a network of dictionaries a challenging task. As in our case for three specific Bantu languages, data is rarely available in a schematic and technical homogeneous way. Transformation into a common model such as the BLM is, therefore, a first helpful step towards aligning datasets in a more straightforward fashion.

Missing reference data is another problematic aspect that has to be dealt with. Dictionaries as compiled by the CBOLD project have been compiled over decades and have only been assigned with loose and inconsistent translations to English or French instead of direct translations to other Bantu languages. By linking lexemes to concepts within WordNet, stable referencing of an external vocabulary can be ensured. This provides a common basis for linking with further dictionary data in the future.

The result of dictionary alignment is a relevant resource for fields such as teaching, where comprehensive dictionaries of high quality that may include references to external and even non-lexical data such as sample sentences or similar words are of fundamental importance. In the context of countries like South Africa, it becomes obvious that there is an urgent need for such data since mother-tongue education has gained popularity in recent years while the importance of international languages such as English is also incorporated into teaching concepts.

To allow for these use cases and an even wider applicability of dictionaries, the overall reliability and consistency of the data need to be assured. The presented systematic extraction and preparation of a shared integration model allows for collaborative approaches to quality assurance which can significantly boost the grade of the data.

Future work will include the incorporation of additional dictionaries based on the BLM and improving and extending their bilingual alignment. Further possibilities for expanding the alignment between dictionary entries in different languages needs to be considered. For the similarity of translations or descriptions in the pivot language English (or French), not only simple string similarities, but also similarities of the corresponding word embeddings can be used to link semantically similar lexemes.

Naturally, meaningful results can only be achieved with direct collaboration with language experts and native speakers. The systematic transformation and enrichment of public dictionaries like the ones provided by CBOLD have the potential to be an important starting point and a valuable resource for the Bantu language family.

References

1. Sonja Bosch, Thomas Eckart, Bettina Klimek, Dirk Goldhahn, and Uwe Quasthoff. Preparation and Usage of Xhosa Lexicographical Data for a Multilingual, Federated Environment. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki (Japan)*, 2018.
2. CBOLD. Bantuists' Manifesto. Website, 1996. Available: `http://www.cbold.ish-lyon.cnrs.fr/Docs/manifesto.html`; Accessed on 8 January 2019.
3. G.-M. De Schryver. *Oxford School Dictionary: Xhosa-English.* Oxford University Press Southern Africa, Cape Town, 2014.
4. Ethnologue. Ndebele. Website, 2019. Available: `https://www.ethnologue.com/language/nde`; Accessed on 8th January 2019.
5. Christiane Fellbaum, editor. *WordNet: an electronic lexical database.* MIT Press, 1998.
6. Jorge Gracia, Marta Villegas, Asuncion Gomez-Perez, and Nuria Bel. The apertium bilingual dictionaries on the web of data. *Semantic Web*, pages 231–240, 2018. `doi:10.3233/SW-170258`.
7. J. T. Mathangwane. *Kalanga.* Comparative Bantu OnLine Dictionary CBOLD, 1994. URL: `http://www.cbold.ish-lyon.cnrs.fr/Load.aspx?Langue=Kalanga&Type=Text&Fichier=Kalanga.Mathangwane1994.txt`.
8. J. T. Mathangwane. *Ikalanga 50 Years On: A Cross Border Language Against Tremendous Odds.* Botswana Notes and Records, 48, 2016.
9. Derek Nurse and Gérard Philippson. *The Bantu Languages.* Routledge, London, 2003.
10. J.N. Pelling. *A Practical Ndebele Dictionary.* Comparative Bantu OnLine Dictionary CBOLD, 1971. URL: `http://www.cbold.ish-lyon.cnrs.fr/Load.aspx?Langue=Ndebele&Type=Text&Fichier=Ndebele.Pelling.1971.txt`.

Cherokee Syllabary Texts: Digital Documentation and Linguistic Description

Jeffrey Bourns
Digital Scholarship Group, Northeastern University, Boston, MA, USA
j.bourns@northeastern.edu

Abstract

The Digital Archive of American Indian Languages Preservation and Perseverance (DAILP) is an innovative language revitalization project that seeks to provide digital infrastructure for the preservation and study of endangered languages among Native American speech communities. The project's initial goal is to publish a digital collection of Cherokee-language documents to serve as the basis for language learning, cultural study, and linguistic research. Its primary texts derive from digitized manuscript images of historical Cherokee Syllabary texts, a written tradition that spans nearly two centuries. Of vital importance to DAILP is the participation and expertise of the Cherokee user community in processing such materials, specifically in Syllabary text transcription, romanization, and translation activities. To support the study and linguistic enrichment of such materials, the project is seeking to develop tools and services for the modeling, annotation, and sharing of DAILP texts and language data.

2012 ACM Subject Classification Applied computing → Digital libraries and archives

Keywords and phrases Cherokee language, Cherokee Syllabary, digital collections, documentary linguistics, linguistic annotation, Linguistic Linked Open Data

Digital Object Identifier 10.4230/OASIcs.LDK.2019.18

Category Short Paper

1 Overview

The Digital Archive of American Indian Languages Preservation and Perseverance (DAILP) is an innovative language revitalization project that seeks to provide digital infrastructure for the preservation and study of endangered languages among Native American speech communities. DAILP is overseen by Northeastern University scholar Ellen Cushman, author of a recent study of the Cherokee Syllabary [2], and supported by the Digital Scholarship Group at Northeastern, the project's host institution [3]. The project's initial goal is to publish a digital collection of Cherokee-language documents to serve as the basis for language learning, cultural study, and linguistic research. Its primary texts derive from digitized manuscript images of documents recorded in the Cherokee Syllabary, a written tradition that spans nearly two centuries. Of vital importance to DAILP is the participation and expertise of Cherokee community members in the transcription, romanization, and translation of these texts. Further enhancements to DAILP texts will include phonemic romanization and free translation layers aligned with the Syllabary text, linguistic annotation, orthographic conversion functionality, parser development, and publication of project datasets as Linguistic Linked Open Data (LLOD). With project infrastructure in place, similar DAILP initiatives are envisioned for Ojibwe and other indigenous languages of North America. This paper describes resources, challenges, and early decisions informing the design and development of the DAILP Cherokee project.


© Jeffrey Bourns;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 18; pp. 18:1–18:6

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Cherokee language and community

The Cherokee language (ISO 639-3, chr) belongs to the Iroquoian language family and survives as the sole representative of the Southern Iroquoian branch. Members of the distantly related Northern Iroquoian branch include Mohawk, Oneida, Onondaga, Seneca, Cayuga, and several further languages now extinct.

A recent report numbers speakers of Cherokee at approximately 12,300 people in the United States, including nearly 10,000 speakers of the Cherokee Nation community in northeastern Oklahoma and 1,000 speakers among the Eastern Band of Cherokee Indians in western North Carolina; to these estimates may be added an undetermined but relatively high percentage of speakers among the 7,500 members of the United Keetoowah Band of Oklahoma and Arkansas [5],[13]. Compared with other Native American languages, Cherokee has a relatively high number of speakers, but the language is spoken by few tribal members under the age of 40, and children at home no longer acquire Cherokee as their first language [12],[15]. Community efforts toward language revitalization include such initiatives as the establishment of Cherokee immersion schools since 2001, yet reversing the language shift will require more robust support for language learning and preservation. Vitality status currently assigned by UNESCO to Oklahoma Cherokee is "definitely endangered," and North Carolina Cherokee is seen as "severely endangered" [11].

3 Cherokee Syllabary and written tradition

Among indigenous languages of North America, Cherokee is notable for its own writing system, the Cherokee Syllabary, and for a written tradition richly documented in this script. The Syllabary was devised in the early 19th century by Sequoyah, a Cherokee silversmith, who introduced the script to tribal leaders in 1821. In the years thereafter the Syllabary was quickly embraced by Cherokee society, which led to widespread literacy and official adoption by the Cherokee Nation in 1825. Compiled over nearly two hundred years, the documentary record of Cherokee Syllabary texts comprises newspapers, almanacs, religious tracts, hymns, laws, pamphlets, private correspondence, and also culturally sensitive materials, such as prayers and magic formulas recorded by traditional Cherokee doctors. Archival collections of Cherokee manuscripts have been preserved and cataloged by such institutions as Yale University and the Smithsonian's National Anthropological Archives (NAA), and with the support of the Cherokee community, recent years have seen Syllabary manuscripts of cultural and historical interest digitized and published online [1].

4 DAILP goals and design

The DAILP initiative builds on digitization of historical Cherokee manuscripts. Under this approach, digitized Cherokee Syllabary documents provide the foundation for multi-layered text collections that can serve the diverse needs and interests of students and scholars of Cherokee language and culture. Project design is guided by the skills and requirements of the Cherokee community itself, particularly as these entail selection and preparation of texts and management of digital access. For gating and access to culturally-sensitive material, the DAILP collection will implement a system of protocols and permissions based on community-defined relationships and requirements. Archival Syllabary texts have been vetted and pre-selected by Cherokee translators for inclusion in the DAILP collection. Among these are numerous handwritten documents of uneven legibility for which automated processing via OCR is impractical. By design, DAILP workflows engage the Cherokee user community

in processing these materials, specifically in Syllabary text transcription, transliteration, and translation activities. Among DAILP's initial goals are the design and development of an interface to support such tasks, informed by the skills and needs of project contributors.

Beyond these basic documentation activities lie more complex processing tasks. A key challenge for DAILP is support for the interpretation and annotation of text editions by contributors of varying levels of literacy and linguistic competence. To language learners and literate readers alike, historical Cherokee texts often pose significant difficulties due to the obscurity of lexical items, the morphological complexity of Cherokee language data, and the variety and ambiguity of Syllabary spellings. To support the interpretation and linguistic enrichment of such materials, the project is seeking to develop tools and services for the lexical and grammatical annotation of DAILP texts. Project editions thus annotated will also serve as a valuable source of primary language data for the development of further descriptive resources for Cherokee. Based on existing well-annotated datasets, recent contributions to Cherokee linguistics, and innovative language data management software, development of such infrastructure is currently underway.

5 DAILP language data

DAILP has acquired and enhanced several datasets of well-structured language data transcribed from descriptive resources for Oklahoma Cherokee. These datasets comprise Syllabary transcriptions, "simple phonetics" transliterations, phonemic representations, grammatical annotations, and English translations. The transcribed lexical data issues from three foundational sources for the study of Cherokee: *Cherokee-English Dictionary* [6], this dictionary's grammatical appendix [14], and *A Handbook of the Cherokee Verb* [7]. The main source is the dictionary, compiled by community linguist Durbin Feeling. Its appended grammatical outline is a rich source of annotated surface forms, and the verb handbook is similarly detailed and useful.

For phonemic representation, the dictionary and appendix use a romanized orthography known as the number system, which introduced a set of superscript numbers for marking Cherokee pitch patterns. Although unconventional, the number system is familiar and important to the community, thus DAILP plans to store and display surface forms transcribed faithfully from these sources in their original orthography. In addition to the number system transcriptions, a further DAILP dataset provides phonemic transcriptions of the same language data using conventional linguistic notation, which is practical for orthographic conversion functionality. Thus, for example, in addition to its Syllabary representation, the form for "I'm helping him" may be displayed as /jisdeliha/ (simple phonetics), /ji^{1}sde^{2}lị^{3}ha/(number system), or/jìisdeelíha/ (phonemic transcription) in the DAILP interface.

Crucially, these descriptive resources provide the project with an internally consistent generalization over Oklahoma Cherokee primary language data. Much in the way of many older manuscript traditions, spellings across historical Syllabary texts do not reflect an established standard. For DAILP's purposes, Syllabary spellings from the Feeling sources offer a practical standard under which orthographic and dialectal variants from DAILP texts may be subsumed. Surface forms in the Feeling sources are moreover linguistically conservative and preserve, e.g., final syllables, which are typically omitted in written sources. Especially valuable are Feeling's precise and consistent representations of vowel length and tonal configurations, which inform an important recent study of tonal behavior in Oklahoma Cherokee (TAOC) [16]. Together with the Feeling datasets, the specification of phonology in TAOC provides the DAILP project with a practical basis for parser development.

6 Linguistic resources for modeling Oklahoma Cherokee

For its modeling and annotation of project language data, DAILP has drawn mainly on two recent contributions to Cherokee linguistics: the systematic survey of phonology in TAOC, and a modern descriptive grammar of broader scope (CRG) [10]. Both TAOC and CRG offer valuable treatments of Oklahoma Cherokee, yet these works differ fundamentally in terms of orthographies, morphological analyses, terminologies, tagsets, and target audiences. A key early challenge for DAILP has been to identify and select from among these resources elements and approaches that are 1) practical for the design and implementation of DAILP tools and services, and 2) accessible and informative to a diverse community of users and contributors working with DAILP texts and language data.

For practical purposes, DAILP has made it a priority to deploy linguistic models and conventions that can straightforwardly support development of project infrastructure. Due to its primary reliance on TAOC for both example data and formulation of parser rewrite rules, DAILP has adopted the orthography, morphological analyses, and tags found in TAOC for the project's underlying representations, grammatical annotations, and specification of (morpho)phonology. In further support of this approach, the DAILP project has been fortunate to acquire a database of underlying lexical roots, stems, and affixes established by linguist Hiroto Uchihara, author of TAOC. By comparison with CRG, it should be noted, TAOC provides more granular morphemic segmentations of underlying forms. Accordingly, IGT examples presented in TAOC typically proceed from a deeper layer of derivation, and thus often require the application of more rules than CRG in order to generate well-formed surface forms. Despite this added complexity, the rigorous specification of phonology in TAOC is a significant windfall to project parser development, and DAILP's modeling decisions and dataset preparation reflect this practical advantage.

Designed for both linguists and language learners, CRG is an important descriptive resource for the study of Oklahoma Cherokee. For DAILP's purposes, the main value of CRG lies in its clear and concise explanations of Cherokee grammar and its many helpful examples. Given the complexity of Cherokee language data, ready access to the definitions and descriptions in CRG will be invaluable to users seeking to interpret and annotate DAILP texts, most practically via external reference to a published linguistic ontology. Ontology development moreover aligns with further interoperability goals of the project, based on best practices for Linked Data modeling and publication of DAILP datasets. Toward this end, DAILP is exploring development of Linguistic Linked Open Data (LLOD) tools for language-specific description of Cherokee, drawing on the domain knowledge of CRG as well as that of TAOC and several further resources. Due to the rich polysynthetic morphology of Cherokee, of particular interest to DAILP are such models as OntoLex and the Multilingual Morpheme Core Ontology (MMoOn Core) for representation of lexical and morphological language data [8].

7 Online Linguistic Database (OLD)

Due to multiple features well suited to the project, DAILP has installed and configured the Online Linguistic Database (OLD) as its language data management software. Created by linguist, developer, and DAILP project member Joel Dunham, the OLD is a program for creating collaborative language documentation web services [4]. The OLD was developed to meet the need for multi-user cross-platform tools for language documentation and analysis, and its software is designed specifically to support collaborative storing, searching, processing, and analyzing of linguistic data. Of special interest to DAILP is the OLD's well-documented

utility in storing and analyzing language data from Blackfoot, a polysynthetic language of North America [4]. Likewise valuable is the OLD's parser development tool, which supports on-the-fly manual annotation based on user adjudication and selection of candidate parses. A further asset to DAILP is the OLD's orthographic converter, which enables users to select from among several familiar orthographies for the display of Cherokee phonemic representations. Project needs are also well served by the web services architecture of the OLD, which can interact seamlessly with the DAILP interface created for text processing by the user community.

8 Conclusion

As the pool of native speakers recedes and language shift encroaches on the Cherokee speech community, a sense of urgency attends the DAILP initiative. Interviewed for a recent article, Cherokee language translators working on NAA manuscripts report that these documents contain words and phrases that they hadn't heard in decades. A source for the same report estimates that nearly a third of lexical items attested in Smithsonian manuscripts are either no longer in current usage or else simply unknown [9]. Language revitalization is essential to the elucidation of historical Syllabary texts and to the discovery and preservation of Cherokee cultural and linguistic heritage. In partnership with the community, DAILP seeks to provide a durable window on this written tradition, and tools to help its linguistic heirs safeguard and illuminate its precious legacy.

References

1 Kilpatrick Collection of Cherokee Manuscripts. `http://transcribe.library.yale.edu/projects/collections/show/2`. Beinecke Library, Yale University.

2 Ellen Cushman. *The Cherokee Syllabary: Writing the People's Perseverance*. University of Oklahoma Press, 2012.

3 Digital Scholarship Group, Northeastern University. `https://dsg.neu.edu/`.

4 Joel Robert William Dunham. *The Online Linguistic Database: software for linguistic fieldwork*. PhD thesis, University of British Columbia, 2014.

5 David M. Eberhard et al. Cherokee. In David M. Eberhard, Gary F.Simons, and Charles D. Fennig, editors, *Ethnologue: Languages of the World*. SIL International, twenty-second edition, 2019.

6 Durbin Feeling. *Cherokee-English Dictionary*. Cherokee Nation of Oklahoma, 1975.

7 Durbin Feeling, Craig Kopris, Jordan Lachler, and Charles van Tuyl. *A Handbook of the Cherokee Verb: A Preliminary Study*. Cherokee National Historical Society, 2003.

8 Bettina Klimek. Proposing an OntoLex-MMoOn Alignment: Towards an interconnection of two linguistic domain models. In *Proceedings of the LDK workshops: OntoLex, TIAD and Challenges for Wordnets*, pages 1–16, 2017.

9 Robert Leopold. Articulating culturally sensitive knowledge online: A Cherokee case study. *Museum Anthropology Review*, 7(1-2):85–104, 2013.

10 Brad Montgomery-Anderson. *Cherokee Reference Grammar*. University of Oklahoma Press, 2015.

11 Christopher Moseley, editor. *Atlas of the World's Languages in Danger*. UNESCO, 3 edition, 2010.

12 Cherokee Nation. Ga-du-gi: A vision for working together to preserve the Cherokee language. Report of a needs assessment survey and a 10-year language revitalization plan. Technical report, Cherokee Nation of Oklahoma, 2003.

13 Endangered Languages Project. Cherokee. `http://www.endangeredlanguages.com/lang/chr`.

14 William Pulte and Durbin Feeling. Outline of Cherokee grammar. In *Cherokee-English Dictionary*, pages 235–354. Cherokee Nation of Oklahoma, 1975.

15 Elizabeth Seay. *Searching for Lost City*. The Lyons Press, 2003.

16 Hiroto Uchihara. *Tone and Accent in Oklahoma Cherokee*. Oxford University Press, 2016.

Metalexicography as Knowledge Graph

David Lindemann
Universität Hildesheim, Germany
david.lindemann@uni-hildesheim.de

Christiane Klaes
Universität Hildesheim, Germany
Georg Eckert Institute for International Textbook Research, Braunschweig, Germany
klaesc@uni-hildesheim.de

Philipp Zumstein
Mannheim University Library, University of Mannheim, Germany
philipp.zumstein@bib.uni-mannheim.de

Abstract

This short paper presents preliminary considerations regarding LexBib, a corpus, bibliography, and domain ontology of Lexicography and Dictionary Research, which is currently being developed at University of Hildesheim. The LexBib project is intended to provide a bibliographic metadata collection made available through an online reference platform. The corresponding full texts are processed with text mining methods for the generation of additional metadata, such as term candidates, topic models, and citations. All LexBib content is represented and also publicly accessible as RDF Linked Open Data. We discuss a data model that includes metadata for publication details and for the text mining results, and that considers relevant standards for an integration into the LOD cloud.

2012 ACM Subject Classification Information systems → Resource Description Framework (RDF); Information systems → Document representation; Information systems → Ontologies; Information systems → Information extraction; Information systems → Web Ontology Language (OWL)

Keywords and phrases Bibliography, Metalexicography, Full Text Collection, E-science Corpus, Text Mining, RDF Data Model

Digital Object Identifier 10.4230/OASIcs.LDK.2019.19

Category Short Paper

Supplement Material `http://euralex.org/publications/lexbib-a-corpus-and-bibliography-of-metalexicographical-publications/`

1 Introduction

Our goal is an online bibliography of Lexicography and Dictionary Research (i. e. metalexicography) that offers hand-validated publication metadata as needed for citations, that represents, if possible, metadata using unambiguous identifiers and that, in addition, is complemented with the output of a Natural Language Processing toolchain applied to the full texts. Items are tagged using nodes of a domain ontology developed in the project; terms extracted from the full texts serve as suggestions for a mapping to the domain ontology. Main considerations regarding the project have been presented in [7].

In this publication, we focus on the data model for LexBib items, its integration into the LOD cloud, and on relevant details of our workflow. In Section 2 we describe how publication metadata and full texts are collected and stored using Zotero, data enrichment and transfer to RDF format. Section 3 addresses the text mining toolchain used for the generation of additional metadata, that are linked to the corresponding bibliographical items. As shown in Fig. 1, an OWL-RDF file is the place where this merging is carried out. In Section 4 we describe the multilingual domain ontology that will be used to describe the full text content with keywords or tags.


© David Lindemann, Christiane Klaes, and Philipp Zumstein;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 19; pp. 19:1–19:8

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

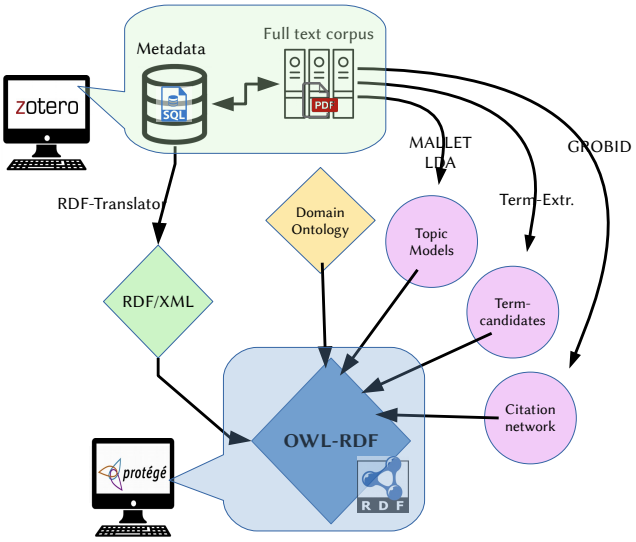


Figure 1 Workflow for combining publication metadata and additional metadata in LexBib.

2 Data Enrichment: Publication Metadata

For the task of web scraping and manually validating publication metadata, and for storing the corresponding full texts, the Zotero software application[1] offers state-of-the-art functions, such as one-click data ingestion from structured metadata as well as from general websites, keyword indexing, attaching of files, notes, and links, and duplicate detection.

In our workflow, a predefined minimal metadata set is collected and hand-validated for every publication, including author(s), title, publishing year, name of the publication (e. g. the journal or the container volume), publication place, etc. This includes the metadata needed for citation, plus (1) the item type, and (2) a unique identifier (DOI, ISBN) for the publication, where available, and/or (3) a URL that leads to the item's landing page on the publishers' web platform, where the original full text can be accessed. The former are all stored as literal strings or integers, the latter two, i.e. DOI/ISBN, and the link to the full text (URL), are stored as Uniform Resource Identifiers (URI).

Zotero stores the metadata in a SQLite relational database, which then can be used for making citations and lists of references (main purpose of a reference management system) but also to export it in various formats. This second point in combination with the easy data extraction from various websites and platforms makes Zotero also interesting as a cataloguing tool for different purposes. For example, Zotero is used as cataloguing and automatic data ingesting tool in the project IndexTheologicus [5]. Moreover, as another example, the one-click option to add a reference in the graphical interface of Wikipedia is supported by Zotero.[2]

[1] See `http://zotero.org`.
[2] See `https://www.mediawiki.org/wiki/Citoid/Zotero%27s_Tech_Talk`

```
<rdf:RDF xmlns:...>
    <bibo:AcademicArticle rdf:about="https://academic.oup.com/ijl/article/25/4/398/923874">
        <bibo:pages>398-436</bibo:pages>
        <bibo:doi>10.1093/ijl/ecs026</bibo:doi>
        <dcterms:language>en</dcterms:language>
        <dcterms:abstract>Corpus-driven lexicography and the International Journal of Lexicography (IJL) made
  their first appearance in the world in 1987 and 1988 respectively. ... </dcterms:abstract>
        <dcterms:title>The Corpus Revolution in Lexicography</dcterms:title>
        <dcterms:creator rdf:nodeID="n5"/>
        <bibo:authorList>
           <rdf:Seq><rdf:li rdf:nodeID="n5"/></rdf:Seq>
        </bibo:authorList>
        <dcterms:isPartOf>
            <bibo:Issue>
                <bibo:volume>25</bibo:volume>
                <bibo:issue>4</bibo:issue>
                <dcterms:date>2012-12-01</dcterms:date>
                <dcterms:isPartOf>
                    <bibo:Journal>
                        <dcterms:title>International Journal of Lexicography</dcterms:title>
                        <bibo:issn>0950-3846</bibo:issn>
                        <bibo:shortTitle>Int J Lexicography</bibo:shortTitle>
                    </bibo:Journal>
                </dcterms:isPartOf>
            </bibo:Issue>
        </dcterms:isPartOf>
    </bibo:AcademicArticle>
    <foaf:Person rdf:nodeID="n5">
       <foaf:givenname>Patrick</foaf:givenname>
       <foaf:surname>Hanks</foaf:surname>
    </foaf:Person>
</rdf:RDF>
```

Figure 2 Sample metadata set exported from Zotero as Bibliontology RDF/XML.

In the LexBib project we use Zotero's Bibliontology RDF/XML translator[3] for exporting to linked data. The name bibliontology comes from the Bibliographic Ontology (BIBO) which is used as the main vocabulary in this translator. Besides BIBO the vocabularies Dublin Core (dcterms), Friend of a Friend (FOAF), and MARC Code List for Relators are mainly used. Some less used item types like software, blog post or audio/video recording are exported by also using vocabularies like DOAP Ontology, Programmes Ontology, SIOC Types Ontology. Finally, there is a special minted namespace within zotero.org which is used for everything which is then still unmapped. The current implementation will be checked and possibly improved by considering also the recently published recommendations for RDF-representation of bibliographic data by the Competence Centre on Interoperable Metadata (short KIM in German) [3].

Publication creators (`dcterms:creator`) in Zotero correspond to the roles performed by persons or organisations, i.e. author, editor, series editor, contributor, translator, and, for reviews, reviewed author. While in Zotero, as for version 5.0, no data field is foreseen for the annotation of persons with identifiers such as ORCID or VIAF IDs, such mapping could nevertheless be done using FOAF element values found in the RDF/XML dump. For this task, in LexBib we propose a collaborative approach: The project team will ensure to find literals that refer to the same person and merge them (e.g. Patrick Hanks and P. Hanks). For each author, a profile page will be created. On dissemination events organized at central conferences of our discipline, and using communication channels used by the community, authors will be asked to attach an ORCID, VIAF or GND identifier to their profile page, along with other useful information, like personal homepages, etc. The advantage of that approach is two-fold: On the one hand, mismatches are avoided, and on the other hand, each person decides whether she wants to display an identifier next to her LexBib records that will link these to any other resource linked to the same identifier.

[3] See `https://github.com/zotero/translators/blob/master/Bibliontology%20RDF.js`.

Publication places are stored as literal strings in the Zotero database, and represented by the Bibliontology translator as `dcterms:publisher` / `address:localityName`. The `localityName` literals can be linked to instances of the GeoNames database, using the GeoNames API and related libraries.[4] [5]

Language names appear in the Zotero publication metadata in the "language" field, which refers to the language a publication is written in, and it is translated to `dcterms:language`. We propose to map all language names to instances of the LEXVO ontology,[6] a resource that contains languages and related information, such as the territory a language is spoken in, alternative names of the language, links to resources like ethnologue, etc. We will repeat the same process for the languages a publication is about (see Section 4). In LexBib, both language of publication and object language are relevant variables in the retrieval of bibliographic items, as filter options. At the same time, the LEXVO integration allows the language names to be displayed according to the users' preferred localisation, and, for example, a retrieval of items that refer to languages spoken in a given country.

3 Additional Metadata

In the LexBib project, computational methods are applied for obtaining term candidates, topic models, and citation references. The results shall be added to the items as additional metadata. Topic weights will be used for ranking bibliographic items with similar full text content. In the following, we explain our approach for generation and modeling of term candidates and citation relations.

3.1 Term Extraction

For term extraction, we use a variant of a tool suite developed at IMS, University of Stuttgart [11, 10], henceforth called "TrEx". It extracts the instances of part-of-speech patterns, e. g. (1) NN (single common nouns), (2) NN-NN (two common nouns), or (3) NN-NN-NN (three adjacent common nouns). Then, it ranks the extracted instances according to their termhood or keyness which is measured by dividing the relative frequency of the instance in a document by the relative frequency of the instance in a reference corpus (weirdness ratio, cf. [1]). We run this method twice for each document; for English,[7] once with the British National Corpus (BNC) as a reference corpus in general language in order to retrieve domain specific terms; and once with the whole LexBib English corpus as a reference corpus in order to identify document specific keywords. An example of term candidates extracted by this approach is shown in [7].

Term candidates will be stored in the LexBib database, linked to the corresponding item. Besides enhancing consistent subject indexing and retrieval, term candidates will be used for a mapping to instances of the LexBib domain ontology (see Section 4). Since we plan to display both term candidates and ontology concepts as metadata for LexBib items,

[4] Accessible at `https://www.geonames.org/`. Libraries for accessing GeoNames API are available at `https://www.geonames.org/export/client-libraries.html`.

[5] A similar mapping would be possible for author affiliation strings, that also can be mapped to places. Affiliations are not part of the standard publication metadata and have to be extracted from the full text, which is a non-trivial task of information extraction, that could be addressed using GROBID (see Section 3.3.); this, however, is not part of the workflow proposed here.

[6] Accessible at `http://www.lexvo.org/`.

[7] Our NLP toolchain in this preliminary stage is set up for English; other languages, starting with German and Spanish, will be considered during the lifetime of the project.

we need our RDF data model to distinguish between those two types of subject headings. Furthermore, by providing provenance metadata we will state, what agent (person, algorithm) generated a content descriptor according to what method (computational toolchain with a certain configuration, set of guidelines for manual validation). For term candidates extracted with TrEx, relevant metadata categories, including provencance, are listed in Table 1. The starting point for our RDF modeling is a proposal presented by German National Library (DNB),[8] that uses the W3C's PROV Data Model and PROV Ontology.[9]

Table 1 Points for provenance data for TrEx iterations and single term candidates.

TrEx run	Term candidate
Source corpus description	TrEx run
Reference corpus description	Weirdness ratio
Retrieved part-of-speech patterns	Term status (manual evaluation)
Weirdness and rank thresholds	Mapping to ontology concept
Timestamp	

3.2 Citation Network

Scientific publications usually contain a reference section at the end. The LOC-DB project [6] developed a software application,[10] that wraps all of the following steps in a single GUI: (1) Optical Character Recognition of the full text item for scanned print publications, (2) the information extraction tools GROBID[11] and ParsCit[12], (3) scripts for queries to external publication metadata collections, and (4) a module for defining and storing citation relations. In the LexBib project we will use this Open Source Software for our text corpus and adapt the steps for our needs.

The GROBID tool works on a plain text version of the PDF full text content (or, if this is not available, on the output of the OCR engine) and isolates the block of bibliographic references, the entries of which are then parsed and converted into a structured format compliant to the TEI guidelines (element `<listBibl>`). GROBID uses Conditional Random Fields (CRF), a supervised machine learning method which learns a model based on annotated training data [9]. Problematic citation styles, i. e. formats that are not properly parsed by the tool, will require further annotated training data. Metadata extracted by GROBID are compared to items found in the LexBib collection,[13] or, if not found, sent to an API of external resources containing OpenCitations, Crossref, and library catalogues such as WorldCat, in order to obtain mapping candidates. Then, one (or several) candidate(s) can be manually chosen and thereby connect the LexBib item to an already online existing item. On the one hand, this mapping is used for updating the `<listBibl>` from the metadata in citation style independent format found in the external source, and for enriching it with URI, as done in LOC-DB project. In addition, we plan to use that output for GROBID's CRF training, and also for updates of the citation relations available at the OpenCitations

[8] See `https://wiki.dnb.de/pages/viewpage.action?pageId=146383331`.
[9] See `https://www.w3.org/TR/prov-dm/` and `https://www.w3.org/TR/prov-o/`.
[10] See `https://github.com/locdb`.
[11] See `https://github.com/kermitt2/grobid`.
[12] See `https://github.com/knmnyn/ParsCit`.
[13] Preliminary experiments related to that are explained in [8].

database.[14] We aim at implementing these features during the duration of the LexBib project.

Based on the extracted references, a citation network is visualised and publication clusters can be identified based on citation relations, as it has been proposed in related work (e.g. [4]). The item relations obtained from the analysis of the reference sections in the full texts include (i) the publications cited in a publication, (ii) the publications citing a publication, and (iii) the membership of a publication in a cluster in a citation network.

4 Domain Ontology

The term "Lexicography" is present in controlled vocabularies used for text content description, such as the Library of Congress Subject Headings (LCSH) or Gemeinsame Normdatei (GND). In these general (i.e., not domain-specific) ontologies, but also in a domain-specific keyword collection, such as the one used for indexing publications at BLLDB,[15] a database for linguistic literature, we find a maximum of one level of hyponym terms linked to that term. However, many relevant concepts in the field of lexicography such as "lemmatization" or "neologism" can already be found in these existing ontologies, along with additional semantic information or even mappings to other vocabularies and classifications, but without a defined relation to the term "Lexicography".

Specific thematic indices of Lexicography and Dictionary Research have been proposed (see [7] for reference), isolated from each other. Most proposals are a flat list of keywords, while some define hierarchical relations between them. It is our aim to create a Domain Ontology for Metalexicography (henceforth, DOME), that consists of a multilingual thesaurus, i.e. a tree-like structure of subject headings, each of which is connected to labels (i.e. lexicalizations) in multiple languages, listing possibly more than one synonym in each language. The root element of this thesaurus, "Lexicography", is linked to the same term in the above mentioned widely used general ontologies. DOME will thus constitute a branch, a further ramification of the latter, adding new concepts but also extending relations between existing concepts; we also plan to map LCSH nodes labeled "Lexicography" that are child elements to languages or disciplines to DOME. In order to provide a highly reusable and interconnected resource, we aim to contribute DOME to various existing infrastructures, such as the Linguistic Linked Open Data Cloud (LLOD), Wikidata, or to the ongoing project coli-conc, a resource for managing and sharing concordances between library knowledge organization systems [2].

Regarding the object language or languages of the contribution, i.e. the language(s) the features presented in the article apply to, LexBib-DOME follows an alternative approach: Instead of having thematic keywords as child elements to language names, as in LCSH or existing metalexicographical keyword indices (cf. [7]), or defining language chapters as dependent to every topic, items will be indexed with the instances in LEXVO that correspond to the object language(s), independently of their thematic classification. As a consequence, DOME avoids redundancy, and bibliographic search queries that combine an object language with a topic can be answered in a straightforward way.

[14] The LOC-DB software output follows an adaptation of the OpenCitations linked data model, cf. `https://opencitations.wordpress.com/2019/01/02/opencitations-enhancement-project-final-report/`.
[15] Accessible at `http://www.blldb-online.de`.

5 Summary and Conclusion

We have presented some details of the data model and workflow proposed for the LexBib project, focusing on aspects that are relevant for the representation and availability of publication metadata as RDF Linked Open Data. An overview of the item relations inside LexBib, and of links to external resources is given in Figure 3.

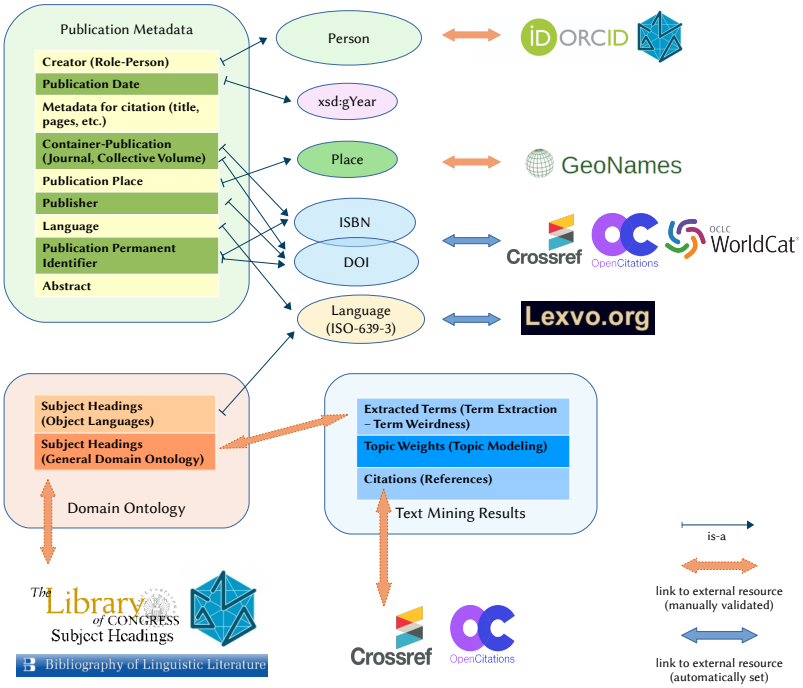


Figure 3 Data Model: Relations inside LexBib and links to external resources.

For the field of Lexicography and Dictionary Research, a domain-specific bibliography with the described features and a thematic index represented as an ontology are an innovation. But we believe that beyond that interest, some questions addressed here are relevant also from a broader or even general perspective.

The LexBib project foresees manual validation and editing effort at several points in the workflow: (i) aggregating and completing the publication metadata and full text collection, (ii) processing and enriching them as linked data, and (iii), the generation of additional content-describing metadata through a combination of computational and manual means. We track and analyse manual work performed for the different tasks as process metadata. This allows then, on the one hand, to evaluate the performance of different combination settings for computational tools and manual validation, and, on the other, to make predictions about the manual work to be foreseen in similar workflows for broader domains.

Regarding LOD integration, we have pointed out for which elements existing vocabularies can be re-used. There are no established standards for the representation of content descriptors, including provenance metadata, as we need it. For us, it is necessary to be able to annotate and, as users of LexBib, to identify content descriptors as, for example, as manually validated keywords that belong to a certain controlled vocabulary, or as term candidates extracted with a certain method, or as set of topic weights relative to a corpus

of publications. With LexBib, we can make a substantial contribution to ongoing work on developing such provenance standards, thus improving transparency and reproducibility of content metadata.

References

1 Khurshid Ahmad, Andrea Davies, Heather Fulford, and Margaret Rogers. What is a term?: The semi-automatic extraction of terms from text. In Mary Snell-Hornby, Franz Pöchhacker, and Klaus Kaindl, editors, *Benjamins Translation Library*, volume 2, page 267. John Benjamins Publishing Company, Amsterdam, 1994. doi:10.1075/btl.2.33ahm.

2 Uma Balakrishnan. DFG-Projekt: Coli-conc. Das Mapping Tool "Cocoda". *o-bib. Das offene Bibliotheksjournal / herausgegeben vom VDB, Bd. 3, Nr. 1 (2016)*, 3(1):11–16, March 2016. doi:10.5282/o-bib/2016H1S11-16.

3 AG KIM Gruppe Titeldaten DINI. *Empfehlungen zur RDF-Repräsentation bibliografischer Daten*. Deutsche Initiative für Netzwerkinformation (DINI), version 2.0 edition, November 2018. URL: https://edoc.hu-berlin.de/handle/18452/2153.3.

4 Nees Jan van Eck and Ludo Waltman. Citation-based clustering of publications using CitNetExplorer and VOSviewer. *Scientometrics*, 111(2):1053–1070, May 2017. doi:10.1007/s11192-017-2300-7.

5 Thimotheus Chang-Whae Kim and Philipp Zumstein. Semiautomatische Katalogisierung und Normdatenverknüpfung mit Zotero im Index Theologicus. *LIBREAS*, 29:47–56, July 2016. doi:10.18452/9093.

6 Anne Lauscher, Kai Eckert, Lukas Galke, Ansgar Scherp, Syed T. R. Rizvi, Sheraz Ahmed, Andreas Dengel, Philipp Zumstein, and Annette Klein. Linked Open Citation Database: Enabling Libraries to Contribute to an Open and Interconnected Citation Graph. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries - JCDL '18*, pages 109–118, Fort Worth, Texas, USA, 2018. ACM Press. doi:10.1145/3197026.3197050.

7 David Lindemann, Fritz Kliche, and Ulrich Heid. Lexbib: A Corpus and Bibliography of Metalexicographcal Publications. In *Proceedings of EURALEX 2018*, pages 699–712, Ljubljana, 2018. URL: http://euralex.org/publications/lexbib-a-corpus-and-bibliography-of-metalexicographical-publications/.

8 David Lindemann, Fritz Kliche, and Kristin Kutzner. Lexikographie: Explizite und implizite Verortung in den Digital Humanities. In Georg Vogeler, editor, *DHd 2018 - Kritik der Digitalen Vernunft, Konferenzabstracts*, pages 257–261, Köln, 2018. Universität zu Köln.

9 Laurent Romary and Patrice Lopez. GROBID - Information Extraction from Scientific Publications. *ERCIM News, Scientific Data Sharing and Re-use*, 100, January 2015. URL: https://hal.inria.fr/hal-01673305/document.

10 Ina Rösiger, Julia Bettinger, Johannes Schäfer, Michael Dorna, and Ulrich Heid. Acquisition of semantic relations between terms: how far can we get with standard NLP tools? In *Proceedings of the 5th International Workshop on Computational Terminology (Computerm2016)*, pages 41–51, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL: http://aclweb.org/anthology/W16-4706.

11 Ina Rösiger, Johannes Schäfer, Tanja George, Simon Tannert, Ulrich Heid, and Michael Dorna. Extracting terms and their relations from German texts: NLP tools for the preparation of raw material for specialized e-dictionaries. In Iztok Kosem, Miloš Jakubíček, Jelena Kallas, and Simon Krek, editors, *Proceedings of the eLex 2015 conference*, Ljubljana; Brighton, 2015. Trojina, Institute for Applied Slovene Studies; Lexical Computing Ltd. URL: https://elex.link/elex2015/conference-proceedings/paper-33/.

Cross-Dictionary Linking at Sense Level with a Double-Layer Classifier

Roser Saurí
Dictionaries Technology Group, Oxford University Press, UK
roser.sauri@oup.com

Louis Mahon
Dictionaries Technology Group, Oxford University Press, UK
Oxford University, UK
louis.mahon@linacre.ox.ac.uk

Irene Russo
Dictionaries Technology Group, Oxford University Press, UK
ILC *A. Zampolli* - CNR, Pisa, Italy
irene.russo@ilc.cnr.it

Mironas Bitinis
Dictionaries Technology Group, Oxford University Press, UK
mkbitinis@gmail.com

Abstract

We present a system for linking dictionaries at the sense level, which is part of a wider programme aiming to extend current lexical resources and to create new ones by automatic means. One of the main challenges of the sense linking task is the existence of non one-to-one mappings among senses. Our system handles this issue by addressing the task as a binary classification problem using standard Machine Learning methods, where each sense pair is classified independently from the others. In addition, it implements a second, statistically-based classification layer to also model the dependence existing among sense pairs, namely, the fact that a sense in one dictionary that is already linked to a sense in the other dictionary has a lower probability of being linked to a further sense. The resulting double-layer classifier achieves global Precision and Recall scores of 0.91 and 0.80, respectively.

2012 ACM Subject Classification Computing methodologies → Lexical semantics; Computing methodologies → Language resources; Computing methodologies → Supervised learning by classification

Keywords and phrases Word sense linking, word sense mapping, lexical translation, lexical resources, language data construction, multilingual data, data integration across languages

Digital Object Identifier 10.4230/OASIcs.LDK.2019.20

Acknowledgements We are very grateful to Charlotte Buxton and Rebecca Juganaru, the expert lexicographers who have contributed all the dictionary knowledge we were lacking and have helped with manual annotations. In addition, we want to express our thanks to Richard Shapiro, Will Hunter and Sophie Wood for their great support in different aspects of the project. All errors and mistakes are responsibility of the authors.

1 Introduction

Dictionary usage has changed tremendously in the past decades, both in terms of quality (e.g., type of searches, preferred support: paper or digital, etc.) and quantity (number of dictionary users, average number of searches by user, etc.). That dictionaries as a product are in decline is a well-known fact, but this trend is not appreciated in the case of bilingual dictionaries. In spite of the availability of free translation tools of remarkable quality, often integrated in web browsers, the generalization of internet access paired with the growth of online content in multiple languages seems to guarantee their continuance.


© Roser Saurí, Louis Mahon, Irene Russo, and Mironas Bitinis;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 20; pp. 20:1–20:16

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

An obvious and very widespread use case for bilingual dictionaries is supporting second language learning. Although learners can nowadays resort to online content for a quick translation, manually edited dictionaries remain the go-to sources for good quality information, especially with respect to less frequent uses, and wider descriptions on how words are employed. This is because dictionaries filter out noisy content, distill the key aspects of linguistic expressions, and provide a broad view on words, e.g., labels for register or domain.

Bilingual dictionaries also play a key role in several language technology areas. For instance, they are a component of search engines for cross-lingual information retrieval, or in metadata tagging tools for multilingual image search systems. Moreover, they are complementary to machine translation systems, which despite their significant improvement with the advent of neural networks technology in the past years, still fall short of returning adequate or informative enough answers when it comes to translating words or lexical constructions provided out of context.

The manual compilation of dictionaries is nevertheless a costly and time-consuming activity, which has led to efforts towards developing methods for (semi-)automating the process. An example of this is the shared task *Translation Inference Across Dictionaries* (TIAD), initiated in 2017 with the aim of exploring methods and techniques to auto-generate bilingual and multilingual dictionaries based on existing ones.[1] The current paper presents research in a similar direction. In particular, it introduces a piece of work embedded within a wider programme with a two-fold goal:

1. Automatically creating new bilingual dictionaries, a task that touches upon the area known as *lexical translation*, concerning systems able to return translations of words or phrases, e.g., [15].
2. Enriching existing bilingual dictionary information with additional data available from other lexical resources (e.g., sense definitions, grammatical notes, domain information, etc.). This second task has to do with the area referred to as *word sense linking* (aka *sense alignment*, *sense mapping* or *sense matching*) [10].

To these ends, we developed a system for linking entry senses from a monolingual dictionary in language L and a bilingual dictionary between languages L and L' whenever they correspond to the same meaning. In particular, we considered sense links between a monolingual English dictionary and the English side of an English-L' bilingual dictionary.

Linking senses from a bilingual dictionary to a monolingual one is the first step towards goal 2 above of enriching the content of bilingual sources, given that monolingual dictionaries tend to offer information of a different nature from that available in bilingual dictionaries. Furthermore, this same sense linking component can feed into a broader system for developing new bilingual dictionaries. By taking the monolingual dictionary as the pivot to which several bilinguals are linked at the sense level, we expect to be able to automate the creation of bilingual dictionaries involving language pairs not covered by the original bilinguals, therefore addressing goal 1 above. The process is illustrated in Figure 1.

Given an initial phase (Step 1) where the senses in the English side of the bilingual dictionaries are linked to the corresponding senses in the English monolingual dictionary, it should be possible to then move to a second phase (Step 2) where the English senses act as the bridge between the non-English parts of the two bilinguals, thus generating a bilingual dictionary for a new language pair. This paper focuses on the work carried out for Step 1.

[1] See: `https://tiad2017.wordpress.com/` and `http://tiad2019.unizar.es`

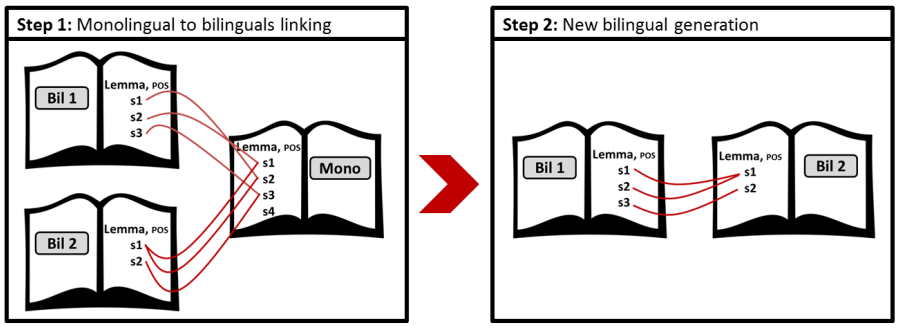


Figure 1 Automated bilingual dictionary generation process.

One of the main challenges of the sense linking task is the fact that it is not restricted to one-to-one mappings. Dictionaries differ in terms of *sense granularity* (that is, one sense in a dictionary corresponds to two or more in another), and in terms of *coverage* (one sense in a dictionary does not correlate to any in the other). Throughout the paper we will refer to this type of misalignment as *non one-to-one mappings*. A further challenge, in this case specific to our project, has to do with the different nature of information in bilinguals as opposed to monolinguals. While the latter tend to contain more extensive textual elements, bilinguals do not have definitions but describe senses by means of translations or short glosses. We will show that the system we put forward here offers a solution to these two issues.

The paper is structured as follows. Section 2 discusses related work. Then, sections 3 and 4 describe the solution proposed to the task at hand. In the first of these sections we give a global overview on the methodology we followed, while the second one goes into the design details of the system we developed. Results are presented in Section 5, and Section 6 closes with final remarks and suggests directions for future work.

2 Related Work

The work presented here belongs to the area of *sense linking* and also, although less directly, to that of *lexical translation*. Less directly in the latter case because, as just argued, the development of a full lexical translation system has yet to be completed. In spite of that, we considered it worth reviewing previous work also on that second area.

Sense linking. The past years have witnessed notable activity in this field, motivated by the interest in developing large Linked Lexical Knowledge Bases (LLKBs) by means of integrating multiple resources into a single one (e.g., BabelNet [17], UBY [9]) in order to achieve maximum lexical coverage and information richness, and thus to be able to better support different NLP tasks. Most of this previous activity involves direct sense linking of Lexical Knowledge Bases (LKBs), as opposed to more traditional dictionary content, even if shaped as Machine Readable Dictionaries (MRDs), e.g., Niemann and Gurevych [18] among many others. The difference between dictionaries (or MRDs) and LKBs is that the latter organize their content in a graph-based structure, depicting the lexical relations that hold among words (e.g., hyper- and hyponymy, entailment, synonymy, etc.). Thus, much of the research on LKB sense linking benefits from lexical information structural organization.

Nevertheless, there is also some work around sense linking which disregards information organization structure and is based solely on similarity between textual elements such as definitions. This approach appears more suited for sense linking dictionary content, although

as will be seen next, in some cases it has been applied to LKBs only. A first strategy here relies on *word overlap*, that is, on the number of words shared by the textual elements in each dictionary, e.g., the early work by Lesk [13] and Byrd [2]. More recently also, Ponzetto and Navigli [19] used word overlap for a conditional probability-based approach for aligning Wordnet and Wikipedia. There are some significant shortcomings of this strategy: it strongly depends on the presence of common words, and in some cases the number of shared words is the same for different senses of the same entry, making the decision hard.

A second, more elaborate strategy consists in representing dictionary textual elements as *vectors in a multi-dimensional vector space* and then computing the distance between them as a proxy for their similarity. The closer the vectors, the more similar the texts they represent. Ruiz-Casado and colleagues [21], for example, followed this strategy for sense aligning Wikipedia articles to their corresponding WordNet synsets [6]. Nevertheless, two major drawbacks of this strategy are, first, the need to set a threshold for determining equivalent senses; and second, the fact that only one-to-one mappings can be accounted for, while it is often the case that a sense in one dictionary corresponds to several in the other.

These issues are not shared by other research resorting to well-known *graph-based methods* for modelling textual information. For example, Ide and Veronis [11] built a complex network of senses and the words present in their definitions, and applied a spreading activation strategy for identifying sense pairs between the *Oxford Advanced Learner's Dictionary* (OALD) and the *Collins English Dictionary* (CED). Although the authors reported good results (90% accuracy), the experiments were unfortunately quite partial as they were applied to only 59 senses selected from OALD. What is more relevant for us here is the fact that the proposed system, seemingly successful when applied to two monolingual dictionaries, does not appear suitable for linking a monolingual and a bilingual dictionary, given that the latter does not contain sense definitions but only translations and indicators.

To our knowledge, there is no work applying a *Machine Learning (ML)* based approach yet to the task of sense linking dictionary content This is the strategy adopted in this project because it can handle non one-to-one mappings and does not require setting any threshold.

Lexical translation. Lexical translation involves systems capable of providing translations for words or lexical expressions. It is closely related to the automatic creation of bilingual dictionary content, especially concerning languages for which there are no translation lexicons of any sort. Work in this area tends to rely on the combination of several bilingual dictionaries to generate a new one involving a language pair not covered in the initial bilingual lexicons. A basic strategy for that is known as *triangulation*. It generates new translation pairs from a source language L_{source} to a target language L_{target} by simultaneously translating from L_{source} to 2 intermediate languages, L_{inter_1} and L_{inter_2}, and from each of these to the target language L_{target}. The final translation is obtained from what is shared in both translation paths. See for example [8, 14].

A second strategy is based on *translation cycles* across languages (as in, e.g., [24, 1]). A cycle is a translation chain across different languages which starts and ends with the same term. For instance, $t_{L_1} > t_{L_2} > t_{L_3} > t_{L_1}$, where t_L is the term used for a word in language L, and $t_L > t_{L'}$ expresses that term t_L translates to $t_{L'}$. Not all translation chains correspond to translation cycles due to the semantic shift that may take place between translations (e.g., a word in one language can have a wider or narrower meaning than its translation in another). Thus, this approach considers as valid only the translation pairs within a translation cycle. Translation cycles tend to give a good precision score because the cycle guarantees translation validity, but low coverage due to its restrictiveness.

Finally, a third approach is based on the notion of *transitivity chains*. That is, the possibility of translating term t_A to term t_C if it is the case that $t_A > t_B$ and at the same time $t_B > t_C$. There are different takes on that, e.g., using probabilistic inference algorithms [15], supporting the decision with parallel corpora [20], or training a machine learning classifier [5].

All these approaches, however, rely exclusively on bilingual dictionaries. This means that a potential lower degree of lexical coverage in any bilingual dictionary used as intermediate step will cause the triangulation or chain to fail. Similarly, differences in sense granularity between two bilinguals may invalidate the linking with a third one. These issues can be avoided if using a more complete, finer-grained monolingual dictionary as a pivot to which to link all bilingual dictionaries. The monolingual dictionary will act as the bridge across the different languages and therefore will ensure consistency on sense equivalence [22, 4, 12, 23, 26]. Our work aligns with this other line of research.

3 Methodology

3.1 General Overview

Since we had a large amount of manually annotated data already available (see Section 3.2), we opted for a ML-based approach. Specifically, we approached the task building a binary classifier capable of judging any sense pair as a *link* (i.e., both senses correspond to the same meaning) or a *non-link* (each sense denotes something different). A sense pair is a pair (s_{mono}, s_{bil}), where s_{mono} is a sense from an entry in the monolingual dictionary and s_{bil} a sense from the same entry in the bilingual dictionary.

The requirement of both senses to belong to the same entry means that they have the same lemma and part of speech (POS) class (e.g., *water* NOUN is different from *water* VERB). We will refer this unit of information as *lexeme*. Note that dictionary homographs (e.g., *lie*$_1$ VERB "*Be in or assume a horizontal position*" vs. *lie*$_2$ VERB "*Tell a lie*") will be considered here as belonging to the same lexeme unit, thus deviating from the standard notion.

Given that the classifier considers each sense pair independently, differences of granularity do not pose a challenge anymore. Any sense in one dictionary can be linked to another sense in the other even if it has previously been linked to a further sense. This strategy, however, is not sensitive to the fact that senses already linked to a sense in the other dictionary have a lower probability of being linked to a second sense. Thus, in order to also benefit from this observation, we complemented the ML classifier with a meta-algorithm which adjusts the judgment on each sense pair based on the potential existence of other links for the same senses in the pair, as will be explained in detail in Section 4.

3.2 Dictionary Sources and Manual Annotation

We took the *Oxford Dictionary of English* (ODE)[2] as the monolingual dictionary, and linked it to the English side of several bilingual dictionaries, also compiled by Oxford University Press, involving English and a second language: English-German (EN-DE), English-Spanish (EN-ES), English-French (EN-FR), English-Italian (EN-IT), English-Russian (EN-RU), and English-Chinese (EN-ZH).[3] To our benefit, the bilingual dictionaries had already been manually linked to ODE at the sense level. The task had been performed by

[2] `https://en.oxforddictionaries.com/` (August 2017 release).
[3] `https://premium.oxforddictionaries.com/`

Table 1 Dictionary fields extracted to build the vector features, by alphabetical order, indicating the type of dictionary they belong to.

Field	Dict. Type	Description
Collocate	Bilingual	Type of words that can be collocated with the word at point (e.g., *food* is a subject collocate for *eat*). Collocates for verbs specify whether they are usually the object or the subject.
Definition	Monolingual	Description of the word meaning.
Domain	Both	Semantic area of a word (e.g., *Medicine*)
Gram. Feature	Both	Grammatical traits of the word. For example, type of complementation pattern for verbs (intransitive, transitive, etc.)
Indicator	Bilingual	Meaning description, generally a one-word or short phrase expression (e.g., *sickly sweet, sweet-tasting*, etc.)
Region	Both	Providing the geographical location of a word (e.g., *British*)
Register	Both	Classifying the tone of a word (e.g., *formal*)
Sense order	Both	Ranking of the sense within its lexeme.

expert lexicographers at Oxford University Press, who examined all senses in the bilingual dictionaries except for those: (a) tagged with the POS classes of *abbreviation* or *symbol*, and (b) presenting no information other than the translation term, i.e., lacking other possible data such as domain, register, region, collocates, example sentences, etc. These annotations were used for training the model and as gold standard to assess results (see Section 3.5).

3.3 Classifier Development Datasets

Instances creation. The dataset of instances for developing our classifier was created as follows: for each lexeme present in both ODE and the bilingual dictionary, we generated all possible sense pairs resulting from coupling each sense s_{mono} from ODE with each sense s_{bil} in the bilingual, i.e., the Cartesian product $S_{mono} \times S_{bil}$, where S_{mono} and S_{bil} are respectively the sets of monolingual and bilingual senses for that lexeme. The resulting set of sense pairs included both sense links and non-links.

Next, for each sense pair in $S_{mono} \times S_{bil}$, the dictionary fields in Table 1 were extracted together with the label *link* or *non-link* that had been manually tagged. Sense pairs for which the bilingual sense had only a translation and no other information, were excluded. The translation field was not useful for our purposes. The extracted pieces of dictionary information were used to build feature vectors, as will be explained in Section 4.1.

Splitting the dataset by POS class. Some POS classes tend to have a higher degree of polysemy than others. Verbs, for instance, are significantly more polysemous than nouns, and even more so than adverbs, as can be seen in Table 2.

Based on this observation, we experimented with separately trained models for different POS classes. We split the training set into 5 subsets, for (a) adjectives, (b) adverbs and prepositions, (c) nouns, (d) verbs, and (d) all the remainder classes (pronouns, determiners, conjunctions, interjections, etc.). The resulting sizes and their class frequencies (links, non-links) are presented in Table 3.

Table 2 Polysemic behaviour by POS class in the *Oxford Dictionary of English*: % of monosemous entries (i.e., single-sense entries), % of entries with 5 or more senses, % of entries with 10 or more senses, and maximum number of senses found in an entry for that POS class.

	% monosemous entries	% entries 5 or more senses	% entries 10 or more senses	max. no. of senses
Adjs	74.4%	2.2%	0.4%	40
Advs & Preps	83.4%	1.7%	0.4%	26
Nouns	76.8%	3.1%	0.6%	53
Verbs	51.2%	11.5%	2.6%	49
Other	72.3%	5.4%	0.7%	22

Table 3 Dataset characteristics: Number of instances, percentage of instances over the dataset, percentage of instances corresponding to links, percentage of instances corresponding to non links.

	No. instances	% instances	% links	% no links
Adjs	228,170	13.7%	37.5%	62.5%
Advs & Preps	42,369	2.5%	35.7%	64.3%
Nouns	824,503	49.6%	31.5%	68.5%
Verbs	556,969	33.5%	15.7%	84.3%
Other	11,256	0.7%	50.6%	49.4%
All POS	1,663,267	100%	27.2%	72.8%

3.4 Building the Classifier

ML classifier. The system features were engineered following recommendations from expert lexicographers from Oxford University Press, who were very acquainted with the content in the different dictionaries. We ran several rounds of experiments and assessed results using standard measures of feature importance and feature ablation techniques. Section 4.1 describes the key features in more detail, while the appendix provides the complete list. We experimented with different ML algorithms (Naïve Bayes, Support Vector Machines, Decision Trees), and based on results opted for the ensemble method Adaboost applied on DTrees.[4]

Meta-classifier. Judging each possible sense pair independently from the others allows to handle the challenges posed by non one-to-one mappings (i.e., differences of granularity and coverage). Nevertheless, sense links are to some extent dependent on the existence of other sense links in the same lexeme. That is, a sense in one dictionary already linked to a sense in the other dictionary has a lower probability of being linked to an additional sense. This observation prompted the development of a meta-classifier sensitive to the number of senses already linked in the same lexeme. We compared results from applying or not applying this algorithm on top of the ML-based classifier. Thus, we investigated two experimental settings:

- **Single-layer classifier:** Using an ML classifier only
- **Double-layer classifier:** Using an ML classifier in combination with the meta-classifier

[4] Specifically, we used python `sk-learn` implementation of [7], with parameters tree maximum depth `max_depth=1`, maximum number of estimators `n_estimators=100`, and `learning_rate=1`.

Baseline classifier. Finally, in order to evaluate the performance of each model, we compared the results against those of a baseline classifier. For each lexeme, the baseline classifier simply links the first monolingual sense to the first bilingual sense, the second to the second, and so on. Formally:

$$B((s_{mono_i}, s_{bil_j})) = 1 \iff i = j \tag{1}$$

3.5 Evaluation

In order to avoid overfitting the model, we applied 10-fold cross validation on the manually annotated data, which thus was used as gold standard against which to assess results. Performance was scored by means of Precision, Recall and its associated F1 measure on sense pairs classified by the model as links. In addition, we used Cohen's Kappa as a way to disregard the effect of correct classifications occurring by chance.

4 Experiment Settings

This section presents the experimental settings in more detail. Specifically, Subsection 4.1 describes the features used by the ML classifier, whereas Subsection 4.2 describes the meta-classifier algorithm applied in conjunction with the ML classifier to take into account possible dependencies among sense pairs.

4.1 ML Classifier Features

In total we considered 120 features, 42 of which were selected for the final classifier. The complete list of the selected features is given in the appendix. Here we explain the rationale applied to create them. We developed two types of features: (a) based on the dictionary fields (presented in Table 1), and (b) based on the entry sense structure, i.e., the ordering of senses within each entry.

4.1.1 Features Based on Dictionary Fields

Domain, register and region. In the dictionaries we used, these three fields can be found qualifying different pieces of information, such as the definition in the monolingual dictionary, the translation in the bilingual, or some example sentences. We extracted domain, register and region elements while differentiating the piece of data they were associated to, and built independent features for each of these. There were 2 types of features based on these fields:

- Boolean features indicating whether the monolingual or bilingual sense has *domain* (or *register*, or *region*) information;
- Similarity scores (ranging [0,1]), comparing the *domain* (or *register*, or *region*) tags from the monolingual and bilingual dictionaries. Similarity was computed in one of two ways: either by the Wu-Palmer metric on WordNet [25], or by measuring how often the two tags cooccurred on the same sense in the same dictionary (note, this value is 1 if and only if both tags are the same).

A single "cross comparison" feature was also included comparing the tags from all possible locations in one dictionary (definition, example sentences, etc.) with the tags from all possible locations in the other.

Indicators and definitions. For each sense pair, the monolingual *definition* and the bilingual *indicator* were compared using two features:

- A Boolean feature indicating if they had a word in common;
- A semantic similarity score (ranging [0,1]) calculated as the cosine similarity between vectors generated with `word2vec` on the GoogleNews corpus, thus leveraging recent advances in word embedding technologies [16] to compute more accurate semantic comparisons.

Grammatical features. We built a Boolean feature for verbs only, encoding if both dictionary senses shared the same complement pattern (i.e., transitive, intransitive, etc.). Similarly, nouns had two bespoke features, signaling if the monolingual and bilingual senses shared the same countability (mass vs. count) and type (proper vs. common).

All textual fields. As a final semantic comparison, we concatenated all text fields from the bilingual sense on the one hand, and all text fields from the monolingual sense on the other, and compared the two resulting text segments using word vectors as described above.

Naive Bayes. One of the major challenges that emerged in this project was the sparsity of each feature. Because there are many possible types of dictionary information for each sense (*domain, register*, etc.) with only one or two actually being realized, the majority of features were null most of the time. The classifier, however, expected to read the same number of features for all instances, so by default it converted null values to 0, negatively impacting on its performance. Consequently, we found that the more common a feature was the more helpful it was observed to be for our performance, and so a natural course of action was to explicitly design a feature to be non-null. With this in mind, we computed a simple probability estimate using a Naive Bayes classifier on all the non-null features for a given instance, where the assumption of independence let us ignore the null values. We discretized each feature into 10 bins, and equated the conditional probabilities with the empirical probability of a link:

$$p(y = 1|x_i = b) = \frac{N_{i,b,1}}{N_{i,b,0} + N_{i,b,1}} \tag{2}$$

where $N_{i,b,c}$ is the number of data points with *ith* feature equal to b, receiving classification c. The product of all such features was then added as an additional feature, $\prod_{i \in F} p_i$, where F is the set of all non-null features. At the cost of an independence assumption, this feature filtered out the noise introduced by the null values. As this Naive Bayes estimate assumes the features are class-conditionally independent, and as this does not fully hold in practice, the product in (2) is often the product of many small values and so it tends to 0. To counteract this, we worked with the geometric mean of all non-null features, instead of the product. Thus, the value for this feature was given by:

$$\sqrt[|F|]{\prod_{i \in F} p_i} \tag{3}$$

4.1.2 Features Based on Entry Sense Structure

Sense frequency. Some senses for a given entry are more frequent than others, and this partially informs how senses are ranked in an entry. That is, more common senses tend to be placed first. Based on that, we inferred an estimate of the frequency of each sense according to its position in the entry, and used the result to form a feature. We assumed

that the frequency of use of a sense is a monotonically decreasing function of its position in the lexeme, and after experimenting with some obvious choices for such a function, we found $f(n) = 1/n + 1/n^2$ to give reasonable frequency estimates when evaluated qualitatively. This function was then normalized for each lexeme (the number of senses in a lexeme is variable so they must be normalized separately). The resulting feature was the absolute difference of the normalized frequency estimates for each of the two senses.

Main sense. The first listed sense in a lexeme can in general be assumed to be the most commonly used and most general. Therefore it was felt that the first sense of each dictionary could more likely (a) be linked to the first sense in the other dictionary, (b) contain multiple links than later senses in the lexeme. To supply this information to the classifier, we included two Boolean features, which indicated whether the bilingual sense and the monolingual sense were the main senses in their respective lexeme.

Single sense. We hypothesized that senses which are the only sense in their lexeme will more likely be linked to at least one sense in the other dictionary. Thus, we included two Boolean features, indicating whether the bilingual sense and the monolingual sense were single senses in their respective lexemes.

4.2 Meta-algorithm for dependent classifications

The ML classifier considers whether a sense pair within a lexeme corresponds to a link individually, without taking into consideration the existence of other links for the same senses in that pair. A complementary solution to this consists in looking at the set of sense pairs of a lexeme as responding to a dependence pattern. More specifically, in considering whether a sense pair corresponds to a link as being partly determined by whether there are other links already present in the same lexeme.

Take as example lexeme L, which has monolingual senses $\{s_{m1}, s_{m2}, s_{m3}, s_{m4}\}$ and bilingual senses $\{s_{b1}, s_{b2}, s_{b3}\}$, and consider the question of whether to assign a link to sense pair (s_{m4}, s_{b1}). If s_{b1} has already been linked to s_{m1}, s_{m2} and s_{m3}, and if in addition s_{m4} has already been linked to s_{b2} and s_{b3}, then it is unlikely that a further link should be added. If, on the other hand, s_{b1} has yet to be linked to any monolingual sense, and s_{m4} has yet to be linked to any bilingual sense, then it is more likely that a new link should be added. This is based on the assumption that in general we should expect each sense in one dictionary to be linked to exactly one sense in the other. Therefore we should require stronger evidence to add a second link than to add a first link, and stronger again to add a third link, etc.

The meta-classifier is designed to make use of this expectation. It applies after the ML classifier in the following manner. In a first step, it takes the confidence score p returned by the ML classifier for each sense pair (s_{mono}, s_{bil}), which it interprets as the probability of a link taking place between senses s_{mono} and s_{bil}. In total, each lexeme L gives rise to $|S_{mono}| \times |S_{bil}|$ such probability estimates. These estimates are re-calibrated using Isotonic regression, as introduced by [3], to adjust for the otherwise unnaturally low variance that arises from Adaboost averaging across all models in the ensemble.[5]

[5] The details of this are beyond the scope of the current paper, but are explained in a general way in the reference provided above.

Then, the meta-algorithm assesses whether each sense pair $(s_{mono}, s_{bil}) \in \{S_{mono} \times S_{bil}\}$ corresponds to a sense link, one at a time and in decreasing order of the estimates p output by the ML classifier. It does this by supplementing the original ML classifier estimate p with two additional probability estimates on whether the sense pair corresponds to a link. These additional estimates are computed using (a) the number of bilingual senses to which s_{mono} has already been linked, and (b) the number of monolingual senses to which s_{bil} has already been linked.

This is done invoking the cumulative distribution $\tilde{F}(x)$, which indicates the probability that a given sense is truly linked to more than x senses in the other dictionary. This function is approximated empirically as:

$$\tilde{F}(x) \approx F(x) = 1/N \times |\{s \in D \mid s \text{ has at most } x \text{ links}\}| \tag{4}$$

where $N = |D|$, that is N is the size of the whole dataset D. The values of $F(x)$ for $0 \leq x \leq 15$ are computed during the pre-processing phase, 15 being the maximum number of observed links for a sense in our dataset. These values are then stored for use by the meta-classifier in order to compute the 2 additional estimates of a link between s_{mono} and s_{bil}: (a) $1-F(m)$, and (b) $1-F(n)$, where m and n are the numbers of links already assigned to s_{mono} and s_{bil}, respectively. These estimates are then combined into a voting ensemble as:

$$(1 - \lambda_1 - \lambda_2)p_{\text{ML}}((s_{mono}, s_{bil})) + \lambda_1(1 - F(m)) + \lambda_2(1 - F(n)) \tag{5}$$

where λ_1, λ_2 are the voting weights and p_{ML} is the probability estimate of the ML classifier.

Just as the vanilla ML classifier assigns a link iff $p_{\text{ML}}((s_{mono}, s_{bil}) = 1) > 0.5$, the meta-classifier assigns a link if and only if the value in (5) is greater than 0.5. Experimentally, we found the best results setting $\lambda_1 = \lambda_2 = .25$. That is, assigning a link if and only if:

$$\begin{aligned} 0.5(p_{\text{ML}}((s_{mono}, s_{bil}))) + 0.25(1 - F(m)) + 0.25(1 - F(n)) > 0.5 \quad &\Rightarrow \\ p_{\text{ML}}((s_{mono}, s_{bil}) = 1) > 0.5(F(m) + F(n)) \end{aligned} \tag{6}$$

Thus, one way to view the action of the meta-classifier is as a method for varying the threshold required for linking, based on the already identified sense links in the same lexeme. The ML classifier classifies a sense pair as a link if and only if its probability estimate p_{ML} exceeds 0.5, while the meta-classifier replaces this global value (i.e., *global* in the sense that it is same for all sense pairs) with a *local* threshold T, which varies for each sense pair depending on the other sense pairs in the same lexeme.[6] Specifically,

$$T = \frac{F(m) + F(n)}{2} \tag{7}$$

If no links have yet been assigned to the senses in a sense pair (i.e., $m = n = 0$), then T will be small and so even a small probability estimate will be sufficient for a positive classification. If by contrast, the senses in question have already been linked to several other senses (i.e., m, n are large), then T will also be large and thus the estimate of the ML classifier will have to be high in order for a positive link to be assigned. The complete action of the meta-classifier is presented in Algorithm 1.

[6] To be more precise, depending on the other sense pairs with a higher p_{ML} estimate, since they will have been previously evaluated as to whether they correspond to a sense link.

Algorithm 1 Meta-classifier algorithm.

for each lexeme L with monolingual senses set S_{mono} and bilingual senses set S_{bil} **do**
  for each sense pair $(s_{mono}, s_{bil}) \in S_{mono} \times S_{bil}$ **do**
    Obtain its probability estimate p_{ML} from the ML classifier
  for each probability estimate p_{ML}, of sense pair (s_{mono}, s_{bil}), from largest to smallest, **do**
    Determine m and n, the number of already existing links for s_{mono} and s_{bil}, respectively
    Compute $T = \frac{F(m)+F(n)}{2}$
    if $p_{\text{ML}} > T$ **then**
      Classify sense pair (s_{mono}, s_{bil}) as a link
    else
      Classify as no link

Though it has only been tested on the present task of sense linking, this algorithm can in theory be generalized to any classification problem in which there is a dependency between the classification on certain sets of elements.[7]

5 Results and Discussion

As just presented, we explored two experimental settings: (a) using only a ML classifier, and (b) applying it in combination with a statistically-based meta-classifier. Table 4 provides the results for the two settings, along with those for the baseline classifier. Performance is evaluated using Precision (P), Recall (R) and its derived F1 score over sense pairs classified as *links*. Our focus was to assess the correctness of what the system had tagged as links (P on links) and its capacity to identify true links (R on links).[8] Moreover, we employed Cohen's kappa as the most common statistic used to account for correct classification that takes place purely by chance. For each metric, for each experimental setting, the best result is in bold face while the worst one is underlined.

Verbs is consistently the worst performing POS class, while the miscellaneous class *Other* performs the best in all cases but one. The good results for *Other* can be explained partly by the fact that it has a perfectly balanced training dataset (see Table 3), and partly by its low degree of polysemy (shown in Table 2), as opposed to verbs, which are the most polysemous POS. *Adverbs & Prepositions* is the second best performing class, also explained by its low degree of polysemy relative to nouns and, more particularly, verbs. Adverbs and prepositions present the highest percentage of monosemous entries, with the number of senses per entry declining very quickly. At most, 26 senses can be found in an entry for an adverb or preposition, which is half the size of the most polysemous entry for nouns.

The level of balance of each dataset (Table 3) is also a factor in the performance for each POS class. This can be appreciated when comparing P&R scores for sense pairs classified as *links* (those reported in Table 4) against P&R scores for sense pairs classified as *non-links*,

[7] For example, the task of assigning tags to YouTube videos could be viewed as a linking task between a set of videos and a set of tags. We might expect each video to be, on average, correctly assigned to around 3 tags. It would be surprising if a video was assigned no tags, or was assigned 20 tags. In the other direction, assuming the given tags were chosen so as to be meaningful and realistic, it would be surprising if one tag was not assigned any videos or if one tag was assigned to every video. These expectations could be leveraged by the meta-classifier. What is described above would require slight modification to work in other domains, but it is reasonable to conjecture that some version of the same idea may prove similarly effective elsewhere.

[8] We also obtained P and R for sense pairs classified as *non-links*, not shown here due to space constraints and given that our interest was on the correctness and coverage of *link*-tagged sense pairs.

Table 4 Performance scores for baseline, ML classifier only, and ML classifier + meta-classifier.

	Precision			Recall			F1			Kappa		
	base line	ML	ML+ Meta	base line	ML	ML+ Meta	base line	ML	ML+ Meta	base line	ML	ML+ Meta
Adjs	0.77	0.91	0.94	0.66	0.83	0.87	0.71	0.87	0.90	0.56	0.79	0.84
Advs-Preps	0.80	0.93	0.94	0.76	0.85	0.90	0.78	0.89	0.92	0.66	0.83	**0.88**
Nouns	0.74	0.89	0.92	0.68	0.78	0.83	0.71	0.83	0.87	0.58	0.76	0.82
Verbs	0.47	0.80	0.83	0.42	0.53	0.64	0.44	0.64	0.72	0.35	0.59	0.67
Other	**0.87**	**0.95**	**0.95**	**0.82**	**0.89**	**0.91**	**0.84**	**0.92**	**0.93**	**0.70**	**0.84**	0.87
All POS	0.70	0.88	0.91	0.63	0.75	0.80	0.66	0.81	0.85	0.54	0.74	0.80

not shown here due to space constraints. The more balanced a dataset, the more similar the P&R values for both types of sense pairs are. By contrast, in the case of verbs (the least balanced class, with only around 16% of links), the difference between the scores for *links* and *non-links* is noticeable. In the double-layer system, P and R for *non-links* respectively raise to 0.94 and 0.98 (vs. 0.83 and 0.64 for *links*). In general, the unbalance in favor of *non-links* results in high R scores for these, ranging between 0.95 and 0.98 across all POS classes. In other words, the system has a stronger tendency to identify sense pairs as *non-links*.

Overall, we assess these results as notably positive. In spite of balance issues, P and R on *links* reach a very decent level of performance. Our interest is on high P over R scores because we prefer correct links at the cost of, possibly, low coverage, which we had initially set at a minimum R score of 0.60. All POS classes attained this target. Similarly, all POS classes except for verbs reached a P score of at least 0.92, which indicates a high degree of correctness. Though not as perfect as hand-curated content, the resulting links (including those for verbs) can already be used for less quality-demanding use cases than traditional dictionaries, such as generating multilingual datasets feeding into cross-lingual search engines or image tagging systems.

Finally, Table 4 shows the positive effect of the meta-classifier. The double-layer system consistently outperforms the ML classifier alone. The improvement is most remarkable for verbs. If classified with the ML classifier only, verbs are 15 points behind *Other* in P and 36 behind in R, but the meta-classifier reduces the gap considerably, a very positive result since verbs correspond to one third of the total data (see Table 3). We thus chose the double-layer setting as our system final design.

6 Conclusions and Next Steps

This paper presented a system for linking senses between a monolingual and a bilingual dictionary. The system approaches the task as a binary classification problem, a strategy which avoids the issue of non one-to-one sense mappings between two dictionaries due to differences in sense granularity and coverage. This classifier was built using Adaboost on Decision Trees and informed with features engineered based on lexicographic knowledge.

Sense links, however, are to some extent dependent on the existence of other links for the same senses. That is, a sense in one dictionary already linked to a sense in the other has a lower probability of being linked to a further sense. Therefore, we experimented with a second classification layer to also model the dependence relation observed among sense links, which was implemented as a statistically based meta-classifier sitting on top of the ML classifier, and which resulted in significantly higher performance scores.

At this point, there are several natural next steps for this project. First, the system can already be used to generate sense links for other monolingual-bilingual dictionary pairs. Second, the double-layer system provides us with a solid framework for developing models for sense linking different types of dictionary pairs (e.g., bilingual-bilingual, monolingual-monolingual, monolingual-thesaurus, etc.), therefore contributing to the creation of a significant linguistic linked data resource. A relevant question to address as part of this work is to what extend the approach adopted here is also applicable to other dictionaries with lower degrees of curation than the ones we used. Last but not least, we can continue work towards our two-fold goal of developing methods for generating new bilingual dictionary content, as well as enriching existing ones with data from linked resources.

References

1. M. Alper. Auto-generating Bilingual Dictionaries: Results of the TIAD-2017 Shared Task Baseline Algorithm. In *Proceedings of the LDK 2017 Workshops, co-located with the 1st Conference on Language, Data and Knowledge*, pages 85–93, 2017.
2. R. J. Byrd. Discovering Relationships among Word Senses. In Antonio Zampolli, Nicoletta Calzolari, and Martha Palmer, editors, *Current Issues in Computational Linguistics: In Honour of Don Walker*, pages 177–189. Springer, Dordrecht, 1994.
3. R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *23rd Int. Conference on Machine Learning*, pages 161–168. ACM, 2006.
4. A. Copestake, T. Briscoe, P. Vossen, A. Ageno, I. Castellón, F. Ribas, G. Rigau, H. Rodríguez, and A. Samiotou. Acquisition of lexical translation relations from MRDs. *Machine Translation*, 9:9–3, 1995.
5. K. Donandt, C. Chiarcos, and M. Ionov. Using Machine Learning for Translation Inference Across Dictionaries. In *Proceedings of the LDK 2017 Workshops*, 2017.
6. C. Fellbaum, editor. *WordNet: an Electronic Lexical Database*. MIT Press, 1998.
7. Y. Freund and R. E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.
8. T. Gollins and M. Sanderson. Improving Cross Language Retrieval with Triangulated Translation. In *24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 90–95. ACM, 2001.
9. I. Gurevych, J. Eckle-Kohler, S. Hartmann, M. Matuschek, C. M. Meyer, and C. Wirth. UBY - A large-scale unified lexical-semantic resource based on LMF. In *Proceeding of the 13th EACL Conference*, pages 580–590, 2012.
10. I. Gurevych, J. Eckle-Kohler, and M. Matuschek. *Linked Lexical Knowledge Bases: Foundations and Applications*. Morgan & Claypool Publishers, 2016.
11. N. M. Ide and J. Véronis. Mapping Dictionaries: A Spreading Activation Approach. In *Proceedings for the New OED Conference*, pages 52–64, 1990.
12. H. Kaji, S. Tamamura, and D. Erdenebat. Automatic Construction of a Japanese-Chinese Dictionary via English. In *LREC 2008*, 2008.
13. M. Lesk. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA, 1986. ACM.
14. G. Massó, P. Lambert, C. Rodríguez-Penagos, and R. Saurí. Generating New LIWC Dictionaries by Triangulation. In R. E. Banchs, F. Silvestri, T. Liu, M. Zhang, S. Gao, and J. Lang, editors, *Information Retrieval Technology*, pages 263–271, 2013.
15. Mausam, S. Soderland, O. Etzioni, D. Weld, M. Skinner, and J. Bilmes. Compiling a Massive, Multilingual Dictionary via Probabilistic Inference. In *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 262–270. ACL, 2009.

16 T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 3111–3119, 2013.

17 R. Navigli and S. P. Ponzetto. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-coverage Multilingual Semantic Network. *Artif. Intel.*, 193:217–250, December 2012.

18 E. Niemann and I. Gurevych. The People's Web Meets Linguistic Knowledge: Automatic Sense Alignment of Wikipedia and Wordnet. In *Ninth International Conference on Computational Semantics*, IWCS '11, pages 205–214. ACL, 2011.

19 S. P. Ponzetto and R. Navigli. Knowledge-rich Word Sense Disambiguation Rivaling Supervised Systems. In *48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1522–1531, 2010.

20 T Proisl, P Heinrich, S Evert, and B Kabashi. Translation Inference across Dictionaries via a Combination of Graph-based Methods and Co-occurrence Stats. In *LDK Workshops*, 2017.

21 M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic Assignment of Wikipedia Encyclopedic Entries to Wordnet Synsets. In *Third International Conference on Advances in Web Intelligence*, AWIC'05, pages 380–386, 2005.

22 K. Tanaka and K. Umemura. Construction of a Bilingual Dictionary Intermediated by a Third Language. In *Proceedings of COLING'94*, pages 297–303, 1994.

23 I. Varga and S. Yokoyama. Bilingual dictionary generation for low-resourced language pairs. In *Proceedings of EMNLP*, pages 862–870, 2009. URL: `http://www.aclweb.org/anthology/D09-1090`.

24 M. Villegas, M. Melero, N. Bel, and J. Gracia. Leveraging RDF Graphs for Crossing Multiple Bilingual Dictionaries. In N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of LREC 2016*, pages 23–28, 2016.

25 Z. Wu and M. Palmer. Verbs Semantics and Lexical Selection. In *32nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, 1994.

26 M. Wushouer, D. Lin, T. Ishida, and K. Hirayama. Pivot-Based Bilingual Dictionary Extraction from Multiple Dictionary Resources. In *PRICAI 2014: Trends in Artificial Intelligence*, pages 221–234, Cham, 2014. Springer International Publishing.

A Features

Feature	Description
bil_dom_direct	Boolean: bilingual domain is non-empty
mono_dom_direct	Boolean: monolingual domain is non-empty
dom_col_sim_avg	co-occurrence similarity score for domain labels, avg if multiple
dom_col_sim_max	as above but max across all values if multiple
dom_col_sim_min	as above but min across all values if multiple
dom_wup_sim_avg	wu-palmer similarity score for domain labels, avg if multiple
dom_wup_sim_max	as above but max across all values if multiple
dom_wup_sim_min	as above but min across all values if multiple
bil_dom_indirect	Boolean: not all the above comparisons are non-empty
dom_cross_comps	a weighted average of the above domain-related features
bil_reg_direct	Boolean: bilingual register is non-empty
mono_reg_direct	Boolean: monolingual register is non-empty, 0 otherwise
reg_col_sim_avg	co-occurrence similarity score for register labels, avg if multiple
reg_col_sim_max	as above but max across all values if multiple

reg_col_sim_min	as above but min across all values if multiple
bil_reg_indirect	Boolean: not all the above comparisons are empty
reg_cross_comps	a weighted average of the above register-related features
bil_ge_direct	Boolean: bilingual regions is non-empty
mono_ge_direct	Boolean: monolingual region is non-empt
ge_col_sim_avg	co-occurrence similarity score for region labels, avg if multiple
ge_col_sim_max	as above but max across all values if multiple
ge_col_sim_min	as above but min across all values if multiple
bil_ge_indirect	Boolean: not all the above comparisons are empty
ge_cross_comps	a weighted average of the above region-related features
bil_ind_direct	Boolean: bilingual sense-level indicator s non-empty
ind_def_wv	cos similarity of sense-level indicators and definition, GoogleNews word vectors
ind_in_def	Boolean, word from sense-level indicators appears in definition
bil_ind_tr_direct	Boolean: bilingual translation-level indicator is non-empty
ind_tr_def_wv	cos similarity of translation-level indicators and definition, GoogleNews word vectors word vectors
ind_tr_in_def	Boolean, word from translation-level indicators appears in definition
bil_ind_ex_direct	Boolean: bilingual example-level indicator is non-empty
ind_ex_def_wv	cos similarity of example-level indicators and definition, GoogleNews word vectors word vectors
ind_ex_in_def	Boolean, word from example-level indicators appears in definition
same_number	Boolean: both marked for same countability (nouns only)
same_trans	Boolean: both marked for same transitivity (verbs only)
same_type	Boolean: both marked for same noun type, (nouns only)
all_text_comp	cos similarity of all the text from one sense with all from the other, GoogleNews word vectors
naive_bayes_estimate	see section 4.1.1
freq	comparison of frequency estimates for each sense, see section 4.1.2
mono_is_main	Boolean: monolingual sense is first in its lexeme
bil_is_main	Boolean: bilingual sense is first in its lexeme
single_sense	Boolean: this sense pair is the only sense pair in its lexeme

Towards the Detection and Formal Representation of Semantic Shifts in Inflectional Morphology

Dagmar Gromann[1]
University of Vienna, Vienna, Austria
https://transvienna.univie.ac.at/en/
dagmar.gromann@gmail.com

Thierry Declerck
DFKI GmbH, Saarbrücken, Germany
ACDH-OEAW, Vienna, Austria
https://www.dfki.de/~declerck/
declerck@dfki.de

Abstract

Semantic shifts caused by derivational morphemes is a common subject of investigation in language modeling, while inflectional morphemes are frequently portrayed as semantically more stable. This study is motivated by the previously established observation that inflectional morphemes can be just as variable as derivational ones. For instance, the English plural "-s" can turn the fabric *silk* into the garments of a jockey, *silks*. While humans know that silk in this sense has no plural, it takes more for machines to arrive at this conclusion. Frequently utilized computational language resources, such as WordNet, or models for representing computational lexicons, like OntoLex-Lemon, have no descriptive mechanism to represent such inflectional semantic shifts. To investigate this phenomenon, we extract word pairs of different grammatical number from WordNet that feature additional senses in the plural and evaluate their distribution in vector space, i.e., pre-trained word2vec and fastText embeddings. We then propose an extension of OntoLex-Lemon to accommodate this phenomenon that we call inflectional morpho-semantic variation to provide a formal representation accessible to algorithms, neural networks, and agents. While the exact scope of the problem is yet to be determined, this first dataset shows that it is not negligible.

2012 ACM Subject Classification Information systems

Keywords and phrases Inflectional morphology, semantic shift, embeddings, formal lexical modeling

Digital Object Identifier 10.4230/OASIcs.LDK.2019.21

Funding Contributions by Thierry Declerck have been supported in part by the H2020 project "ELEXIS" with Grant Agreement number 731015 and by the H2020 project "Prêt-à-LLOD" with Grant Agreement number 825182.

Acknowledgements We would like to thank the anonymous reviewers for their very helpful comments on the original submission of this paper.

1 Introduction

Inflectional morphemes, such as plural *-s* for English nouns, are considered to cause changes in grammatical category without affecting a word's semantics [6]. Semantic shifts are commonly investigated for derivational morphemes, such as *-ment*, that form new lexical items [10], but less so for inflectional morphemes. This study is motivated by the observation that irregularities in inflectional morphemes affect semantic change, a phenomenon that is quite common as we try to show by generating a dataset for the English plural. A monomorphemic example for this phenomenon is the shift from *people* as a common plural of *person* to *peoples*, which refers to a body of persons united by race, ethnicity, and community rather than the

[1] corresponding author


© Dagmar Gromann and Thierry Declerck;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 21; pp. 21:1–21:15

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

plural of people. More complex examples include multimorphemic words, e.g. *pyrotechnics*, as much as phrases, such as *blue devil* the weed as opposed to *blue devils*, which refers to depression. While our analysis focuses on English plurality, other examples such as gender, e.g. *la cabeza* for the physical head in Spanish as opposed to *el cabeza* for a male in charge in Spanish, or tense, e.g. *live* as opposed to the augmented senses of making a *living*, support the argument that this phenomenon generalizes across languages and inflectional morphemes.

Assuming regularities in inflectional morphology may lead to a restrictive view in creating language resources and models. In terms of models, Avrahaman and Goldberg [1], for instance, notice a drop in semantic performance when focusing exclusively on base forms without the words' inflections in training morphological embeddings and attribute this to potential inflectional irregularities (i.e. gender) without further investigating the phenomenon. Their overall recommendation, nevertheless, is to train morphological embeddings on lemmas rather than surface forms. When it comes to popular language resources, WordNet [8] implicitly acknowledges this phenomenon by attributing separate definitions to plural forms of nouns where the meaning changes with the grammatical number. We say "implicit" because when searching for the singular form, e.g. *silk* in the sense of the fabric, the plural and its separate meaning(s) are not available. One has to explicitly search for and be aware of a separate entry for the plural form *silks* to find out that it may refer to garments of a jockey.

To systematically investigate this phenomenon of regular inflectional morphemes that cause semantic shifts, which we call inflectional morpho-semantic variation, we limit our analysis for this paper to grammatical number in English nouns. The proposed method consists in detecting, analyzing, and representing such variants. First, we detect morpho-semantic variants in WordNet based on an augmented number of synsets when querying the plural form of a word. Senses specific to the plural are considered to indicate semantic shift. Second, the context of singular-plural pairs with different senses is evaluated by exploring the distribution of their representation in vector space. To this end, we utilized two pretrained embedding repositories: word2vec [15] and fastText [4]. Results thereof confirm a general intuition of two main types of variants: (i) semantic shifts where plural meanings entirely drift from the singular and lose all connections to its senses, and (ii) those with a clear connection to the singular but also additional meanings. To facilitate an improved representation in computational resources, we propose an extension of the OntoLex-Lemon computational resource[2] that represents linguistic and lexical knowledge in relation to a formal representation in order to accommodate inflectional morpho-semantic variation.

We see our main contributions to the broader topic of language, data and its representation as knowledge as providing:

- a dataset of inflectional morpho-semantic variants of grammatical number,
- a theoretical analysis of different types of such variants,
- an analysis of their representation in conventional vector space, and
- a formal representation method to differentiate inflectional morphemes with and without semantic shift.

To detail these contributions, we first discuss differences in inflection and derivation as well as types of inflectional morpho-semantic variants detected. Section 3 then describes our analysis of such variants in WordNet and our method for extracting a dataset from it as well as the results of that extraction process. Section 4 details the analysis of extracted variants in vector space, followed by a proposal to formally represent them. Prior to some concluding remarks, we discuss approaches related to the analysis of morphology in vector space.

[2] OntoLex-Lemon is the result of a W3C Community Group on representing rich linguistic grounding for ontologies. The final specifications of this model are available at `https://www.w3.org/2016/05/ontolex/`. See also [13] or [14].

2 Inflectional Morpho-Semantic Variants

Morphology investigates the structure of words by analyzing the smallest meaning bearing unit in language, called *morphemes* and their contribution to establishing relations between different words. Morphology most commonly differentiates inflection and derivation. In this paper, we are particularly interested in variants and irregularities of the former type.

2.1 Inflection and Derivation

Inflectional morphology is a set of processes that outwardly change the syntactic information of the word typically without changing its semantics, such as verb tenses. In contrast, in derivational morphology the word form change causes a semantic shift in meaning, such as the affix *un-* in English. Thus, affix patterns attributed to the former category are considered semantically regular with the base form of the respective morphological variants, whereas patterns of the latter category are considered semantically irregular leading to changes in meaning. In other words, while the boundary between these two tends to be a continuum rather than a divide, a generally accepted understanding is that derivation in contrast to inflection changes meaning [6].

2.2 Inflectional Variants

Inflectional morphemes have been studied extensively, however, questions regarding their universality across languages and the exact nature of their semantics remain open [10]. Our interest in this paper focuses on semantics of nominal expressions of grammatical number. In general, it can be stated that singulars denote atomic entities, dual numbers denote pairs, and plurals refer to groups of two or more entities. However, plurals with an associative meaning – denoting one person explicitly and a contextually relevant entity or group such as *los reyes* in Spanish that can denote king and queen – already require a different semantic [10]. In associative meanings and other exceptional cases of grammatical number discussed in Kiparsky and Tonhauser's extensive analysis of inflectional semantics [10], a tight semantic coupling between singular and plural is maintained. In this paper, we are interested in cases were this relation is entirely broken apart and entirely different semantics are assigned to the plural. This has to be differentiated from phenomena such as suppletion [5], where inflection causes drastic changes to the surface form, such as *person* being changed to *people*.

Analyzing grammatical number of English nouns, we noted that in some exceptional cases addition of a plural suffix entirely changes the semantics of the word. For instance, *bloomer* refers to a flower that blooms in a certain way or a loaf of white bread, whereas *bloomers* informally and historically refers to a woman's underpants. Singular and plural share no present-day semantics, even though they might be etymologically related[3]. In such cases the plural exclusively refers to this one meaning without any relation to senses of the singular counterpart of the same word. In other words, semantics of the inflected form have no relation to the semantics of the non-inflected, lemmatized form. As a second major group, plural examples with an idiosyncratic meaning might still simultaneously represent two or more specimen of a specific thing or living being. For instance, *names* refers to name calling in the sense of verbal abuse as much as to two or more designations of things or beings. At

[3] Several sources attribute the plural to the person Amelia Bloomer, a women's right advocate who came to be associated with the clothing reform. However, this does not entirely exclude the possibility of a relation to the singular form of the word.

the same time, the singular *name* might refer to a person's reputation, in which case it might only be used in singular. As an example of living beings in this category, *clams* denote bits of sweet chocolate used as ice cream topping as much as several marine mollusks. In some cases, senses of singular and plural forms might not be identical but share some common characteristics. For instance, *antipode* refers to direct opposites whereas *antipodes* refers to places or regions on diametrically opposite sides of the Earth. From this example we can see that even though the latter meaning is considerably more specific than the former, there are some common traits. We call all of these cases of regular inflectional morphemes that cause semantic shifts morpho-semantic variants and in this paper focus on English nominal constructs of grammatical number, but are confident that other cases, such as gender, would be worth investigating in future.

With the proposed dataset of examples, analysis of word embeddings, and theoretical discussion of the problem, this paper contributes to the investigation of the semantics of grammatical numbers. While the exact scope of the problem of inflectional morpho-semantic variants has not yet been fully identified, an initial dataset extracted from a general language lexicon with an exclusive focus on grammatical number already yielded a significant number of examples (see below). Number and quality of obtained examples suggest that inflectional semantics can be as variable as derivational semantics and are presented in the following.

3 Inflectional Suffixes of Number in WordNet

WordNet [8] has been a very influential lexical-semantic resource over the last decades which is being used in a variety of language technology tasks (e.g. [2]). Its popularity can partially be attributed to its considerable coverage and the quality of mainly manually curated entries. In this section, we investigate the representation of nominal inflectional suffixes of grammatical number in English. Building on this analysis, we propose an approach for the automated extraction of irregular cases for the evaluation of their representation in vector space that is proposed in the next section. To the best of our knowledge, existing datasets of semantic relations in inflectional morphology focus on regular cases. Thus, we think that this dataset can also provide a more encompassing and powerful test bed for models of morphological semantics and its variants. For now, it only looks at English nouns, however, considerable extensions in grammatical category and languages covered are ongoing.

3.1 Representation in WordNet

We observe that WordNet's representation of semantics of words does not inherently support the representation of inflected forms. Semantically similar words are grouped into sets of synonymous words, so-called synsets. When searching for a word in a WordNet interface[4], all potential synsets for this word are returned, including its gloss (a short definition) and all synonyms of the word pertaining to the same synset. While it is possible to actively search for plural forms of a noun, in a vast majority of cases the interface returns results for its uninflected counterpart because it lemmatizes the input. In cases of complementary plural entries, WordNet displays augmented lists of synsets: those associated with the singular, e.g. *people*, and those associated with the plural, e.g. *peoples*. All senses for this example are displayed in Listing 1.

[4] See `http://wordnetweb.princeton.edu/perl/webwn` for a Web interace and `http://www.nltk.org/howto/wordnet.html` for a Python interface integrated in the Natural Language Toolkit (NLTK) [3]

Listing 1 The Synsets for "people" vs. "peoples".

```
people.n.01      ((plural) any group of human beings ... collectively)
citizenry.n.01   (the body of citizens of a state or country)
people.n.03      (members of a family line)
multitude.n.03   (the common people generally)
peoples.n.01     (the human beings of a particular nation or community
                 or ethnic group)
```

This differentiation of grammatical number in the representation of synsets and associated meanings intuitively suggests that plural and singular forms do not share all meanings. Regular cases, such as *car* returns no additional synsets and senses for its inflected form *cars*. Thus, it can be assumed that the change of grammatical number does not cause any semantic shift in those cases. This means, in turn, that it can be assumed that the availability of additional senses indicates such semantic shifts and therefore irregular inflectional forms. When we follow this line of thought for the above example and consult an additional resources[5], we find a clear distinction in meaning between *people*, which itself has to be treated as plural in most senses, and *peoples* with an identical meaning as the corresponding synset in Listing 1. We also find indications that *people* by suppletion is the predominantly used plural of *person* (rather than *persons*). To systematically analyze these irregularities we generate a dataset from WordNet described in Section 3.2.

3.2 Dataset Creation Method

Building on this analysis of joint representations of grammatical number and senses in WordNet, we extract a full list of all available English lemmas in WordNet. Each entry in this list is automatically inflected to its potential plural form, which, if available in the noun list of lemmas, is used to query for its senses. If a query for an inflected form returns senses different from the ones obtained by querying the singular, we consider it an indication of semantic shift.

For a focused analysis of one specific phenomenon of inflectional irregularities, we limit the number of investigated cases to nouns and grammatical number. We analyze English inflectional suffixes for plural nouns, which in our dataset turn out to be: addition of *-s*, addition of *-es*, replacement of *-y* by -ies, replacement of *-us* by *-i*. All potential English nouns are inflected using the *inflect*[6] package in Python and then used to query for WordNet senses in its NLTK corpus.

We implement some restrictions to enhance the quality of the obtained variants. First, we limit the part of speech tag to nouns in order to ensure that all returned senses relate to singular and plural nouns only. Second, only words with at least one overlapping sense in singular and plural are considered. This is due to the fact that WordNet automatically lemmatizes the query word and returns singular and plural senses, the latter only where available, the former even if unrelated to the plural. For instance, *silk* and *silks* share no meanings but the query for the latter still returns all senses of the former. In cases where no singular senses are included due to the lemmatized plural, the words are of a different lemma, such as the personal pronouns *us* and *I* or *faro* the card game and *Faroes* referring to the

[5] We find this same distinction in Merriam Webster's online dictionary `https://www.merriam-webster.com/dictionary/people`

[6] `https://pypi.org/project/inflect/`

island. This is also the reason why we have to subtract all singular senses from the plural to ensure we are left with senses unique to the plural version of the noun. Finally, we remove all senses with lemma names that start with a capital letter to avoid including proper names as plurals, such as *sills* referring to the US operatic soprano Beverly Sills as a plural of *sill*.

In terms of evaluation, all authors of this paper manually checked each resulting singular-plural pair and their senses. Even though we used a morphological analysis tool, several basic grammar rules were violated, such as adding an *-s* to words ending in *-s*, which, for instance, turns the *bos*, a cattle, into the *boss*, the leader. In the formation of *-s* plurals, abbreviations (*aids* related to *aid*), Greek letters (*mu* related to *mus*), chemical elements (*co* related to *coes*), and currencies (*lats* related to *lat*) marked the majority of unreasonable pairings. All of these cases were removed from the final dataset, which lead to 23 removals for the cases of *-s* plural endings and 9 removals for *-es* additions and none for the other two types of endings. We publish a file with all removed entries alongside the actual dataset. The following section represents the resulting dataset that does not take these removed elements into consideration.

3.3 Dataset Results

Applying the described method leads to a dataset of inflectional suffixes for grammatical number that cause semantic shifts. The dataset is published[7] in two versions: (i) one with a header line indicating the type of suffix and a word pair per line of format "singular plural", and a (ii) second version with the same as in (i) and additionally all definitions for the singular and plural from WordNet alongside the synset identifier. Version (i) allows for faster parsing of the variants while version (ii) allows for a detailed tracking and (manual) evaluation of the results. We additionally add evaluations of cosine similarities in vector space to the data repository of this paper.

Table 1 Statistics on Final Dataset.

Suffix type	Number of Examples	Example
-s	419	*silk silks*
-es	11	*rich riches*
-y to *-ies*	24	*fifty fifties*
-us to *-i*	1	*fungus fungi*
All	455	–

Quantified results of the dataset are presented in Table 1 as well as examples for each type. The majority of examples could be detected for the most common additive suffix, while the other suffixes where less common. The table presents one example for each category of morpho-semantic variant. As defined before *silk* denotes the fabric and *silks* a jockey's garments. Second, *rich* conventionally refers to people in possession of wealth, whereas *riches* is commonly used to denote wealth as such. Third, the number *fifty* has to be differentiated from the historical decade *fifties* as well as the time in life between the age of 50 and 60. Finally, *fungus* may refer to an organism, while *fungi* refers to the taxonomic kingdom. We were interested to see in how far word embeddings purely trained on contexts are able to capture these semantic irregularities in inflectional morphemes.

[7] `https://github.com/dgromann/imsev`

4 Inflectional Suffixes of Number in Vector Space

Distributed semantic models capture word meaning purely based on its contexts. Real-valued vector representations of words are obtained by analyzing words occurring in the same sentences, where the window size determines the number of words to the left and to the right that are considered during training with a neural network. Resulting word embeddings have turned out to be highly powerful representations of different semantic aspects of individual words. In our case they are utilized to test whether a purely context-based approach is capable of capturing morpho-semantic variation in grammatical number.

To this end, we utilized two different pre-trained embedding repositories for English: word2vec [15] trained on the Google news corpus and fastText [4] trained on the English web-crawl and Wikipedia corpus[8]. In training, word2vec represents a feedforward neural network with a softmax output layer that trains embeddings with negative sampling, predicting the context for a given center word in its widely used skipgram version. This training model is adapted by fastText to encode character n-grams, where word vectors represent compositions of character n-gram vectors. This has the advantage of a reduced out of vocabulary rate due to the flexible composition of new words based on their n-grams. We decided against the utilization of morphological embeddings such as the ones proposed by Avraham and Goldberg [1] who adapt fastText to combine lemma, surface form, and morphological tag. Both lemma and morphological tag could bias the learned vector space towards ignoring irregularities and thus are counter-intuitive training methods for our purposes.

In order to test the location of a vector we need to navigate through vector space created by the embeddings and analyze the environment of a desired target vector. This can be achieved by querying nearest neighbors of the singular and plural of each word in our dataset (if represented in the vocabulary) and then estimate the overlap of neighbors in the top ten returned closest vectors. Apart from the fact that people seem to frequently misspell *people*, Listing 2 shows that *peoples* is not in the immediate neighborhood.

Listing 2 Top six nearest neighbors of "people" in word2vec.

```
peole: 0.6058608293533325
poeple: 0.59071284532547
individuals: 0.5827618837356567
folks: 0.5794459581375122
peple: 0.578874409198761
peo_ple: 0.5768002271652222
```

Listing 3 displays the same query for words having similar contexts as the word *people* in fastText. We can observe that we get a different list of words, but in both cases the word *peoples* is not included. This is an indication that both words do not share a meaning.

Listing 3 Top six nearest neighbors of "people" in fastText.

```
...people: 0.7241666316986084
'people: 0.6962485313415527
people's: 0.6582629680633545
,people: 0.6566357612609863
''people: 0.6466725468635559
@people: 0.6439790725708008
```

[8] https://fasttext.cc/docs/en/english-vectors.html

Table 2 Evaluation of top ten neighbors of singulars for relation to semantically shifted plurals.

	in top ten neighbors	shared meaning	not in neighbors	plural only	OOV
word2vec	258	237	145	16	53
fastText	303	288	114	19	39

In contrast to Listing 2, Listing 4 shows that *peoples* seems easier to spell and very clearly refers to a very different sense. The singular version with a significantly different semantics does not occur in the list of neighbors. We display here only the word2vec based listing.

Listing 4 Top six nearest neighbors of "peoples" in word2vec.

```
Indigenous_peoples, Similarity: 0.54
Diasporas, Similarity: 0.54
indigenous_peoples, Similarity: 0.53
human_being, Similarity: 0.53
pluralistic_societies, Similarity: 0.53
humankind, Similarity: 0.52
```

For the above case and the represented six senses, an overlap of zero neighbors would be the result, showing a strong indication for a semantic shift that is also captured by the distributed semantic model. We repeated the above experiment with all examples in our dataset and found that embeddings can be utilized in order to neatly separate inflectional morpho-semantic variants that share a meaning with the singular from those without a shared meaning. This can be achieved by evaluating whether the inflected plural form from our dataset is part of the list of ten nearest neighbors.

Querying a plural in WordNet always results in the listing of all singular senses of a word and, where available, senses specific to the plural. However, this rigorous listing of singular senses also applies to plural nouns that share no sense with their singular counterpart. For instance, querying the pants *khakis* would result in a listing of all senses related to *khaki* and that of the plural. Thus, we had to turn to a different resource to obtain the differentiation for plural forms that share senses with the singular and those that do not share any senses, i.e. exist only in the plural version. To this end, Wiktionary usefully differentiates between "plural of" a certain singular word and "plural only". All plural instances in the latter category are considered not to share a meaning with their singular counterparts. As a result, we obtain 412 singular-plural pairs that share meanings and 43 plural words that have no Wiktionary link to a potential singular form. This split helped us evaluate whether word embeddings captured this information since their creation is purely based on context.

As represented in in Table 2, in word2vec 258 plurals are found in the top ten nearest neighbors of their singular of which 237 share senses according to our Wiktionary statistics. For fastext, out of 303 plurals in the top ten neighbors, 288 also share senses according to Wiktionary. However, little overlap could be observed between plurals that are not in the vector neighborhood and have exclusively "plural only" meanings in Wiktionary. We believe that this discrepancy can be attributed to words that are predominantly used in their "plural only" meaning but also share a sense with their singular counterpart. Out of vocabulary (OOV) examples are lower with fastText due its character n-gram encoding method.

While this behaviour is also reflected in the similarity measure between singular and plural, there is no exact division line. Around the values of 0.40 to 0.48 cosine distance pairs in word2vec can belong to either category. Lower values clearly separate input pairs

Table 3 Number of examples per cosine similarity range times 100.

	0-20	20-40	40-60	60-80	80+
word2vec	13	66	129	168	14
fastText	3	27	98	231	54

by meaning, while higher values are good indicators of shared as well as separate meanings. For an overview, we provide ranges of cosine similarity in Table 3, which clearly shows a predominant accumulation of pairs in the range of 0.6 to 0.8 cosine similarity. This preliminary evaluation of the representation of inflectional morpho-semantic variants in vector space needs to be grounded in a more substantial and formal evaluation with several annotators and several evaluation metrics, which we intend to do as part of our future work. Furthermore, we intend to extend the created dataset from other sources with more substantial annotations of inflectional behaviors. Nevertheless, this initial method provides an estimation of the magnitude of each subtypes of inflectional morpho-semantic variants, one where the plural sense is entirely shifted to have nothing in common with the singular and one with partial shifts.

5 Representation in OntoLex-Lemon

The OntoLex-Lemon model was originally developed with the aim to provide a rich linguistic grounding for ontologies, meaning that the natural language expressions used in the description of ontology elements are equipped with an extensive linguistic description. This rich linguistic grounding includes the representation of morphological and syntactic properties of lexical entries as well as the syntax-semantics interface, i.e. the meaning of these lexical entries with respect to an ontology or to specialized vocabularies. The main organizing unit for those linguistic descriptions is the lexical entry, which enables the representation of morphological patterns for each word and/or affix. The connection of a lexical entry to an ontological entity is marked mainly by the `denotes` property or is mediated by the `LexicalSense` or the `LexicalConcept` properties, as this is represented in Figure 1, which displays the core module of the model.

The OntoLex-Lemon model describes at its core an entry-sense relation. Form variants of an entry are encoded as instances of the class `Form` and none of this form variants can be linked directly to a lexical sense, which would be a direct way to represent morpho-semantic phenomena. Therefore, in OntoLex-Lemon morpho-semantic variants can only be represented via their linking to distinct lexical entries.

Our OntoLex-Lemon compliant approach consists in creating a new lexical entry for the plural form that has a specific meaning. We showcase this approach with the word pairs *letter-letters*. While several senses can be associated with both the singular and the plural form of the lexical entry *letter*, the literary culture sense can be associated with the plural form. On the other hand, the sense of literal interpretation (e.g. in the case of law texts that are interpreted by the *letter*) is generally assigned to the singular form. In the following listings, we show, in a simplified manner, the way this complex information can be encoded in OntoLex-Lemon.

Listing 5 displays the lexical entry for *letter*. It is stated that two forms are associated with this noun: a singular (the `canonicalForm`) and a plural (the `otherForm`) form. In this simplified entry, we link only to one sense: the one of an exchange between two parties (see listing 8).

Listing 5 The lexical entry for *letter*.

```
:letter
  rdf:type ontolex:Word ;
  lexinfo:partOfSpeech lexinfo:noun ;
  ontolex:canonicalForm :Form_letter ;
  ontolex:otherForm :Form_letters ;
  ontolex:sense :LexicalSense_letter_1 ;
.
```

Listings 6 and 7 display the basic encoding for the two possible word forms for the entry *letter*.

Listing 6 The form for *letter* in singular.

```
:Form_letter
  rdf:type ontolex:Form ;
  lexinfo:number lexinfo:singular ;
  ontolex:writtenRep "letter"@en ;
.
```

Listing 7 The form for *letters* in plural.

```
:Form_letters
  rdf:type ontolex:Form ;
  lexinfo:number lexinfo:plural ;
  ontolex:writtenRep "letters"@en ;
.
```

The next listing is about the shared sense associated with the lexical entry. As there is a Wikidata entry for the type of entity this sense can refer to, we make use of the `ontolex:reference` property in order to link to this data source.

Listing 8 The lexical sense for the entry *letter* (which can have singular and plural forms).

```
:LexicalSense_letter_1
  rdf:type ontolex:LexicalSense ;
  rdfs:comment "letter as a missive from one party to another (taken
      from Wikidata)" ;
  ontolex:isSenseOf :letter ;
  ontolex:reference <https://www.wikidata.org/wiki/Q133492> ;
.
```

Listing 9 is introducing the additional lexical entry for the plural form of *letter* that has a specific meaning that can not be associated to its singular form. Therefore we link this entry only to the plural instance of the class `Form` and to the specific sense encoded in listing 10, where we additionally formulate the constraint that the usage of this sense is restricted to the plural form *letters*.

Listing 9 The special lexical entry for *letters*.

```
:letters
  rdf:type ontolex:Word ;
  lexinfo:partOfSpeech lexinfo:noun ;
  rdfs:comment "encoding singular and plural entries" ;
  ontolex:canonicalForm :Form_letters ;
  ontolex:sense :LexicalSense_letters_1 ;
.
```

Listing 10 The sense for *letters* in plural.

```
:LexicalSense_letters_1
  rdf:type ontolex:LexicalSense ;
  rdfs:comment "\"letters\" as \"literary culture\"" ;
  ontolex:usage :Form_letters ;
.
```

In fact the use of the `ontolex:usage` property could suffice in order to mark that a sense is restricted to a particular inflectional form of an entry, as exemplified below in Listing 11 for the sense of the literal interpretation, without the need to introduce a new lexical entry.

Listing 11 The literal interpretation sense for *letter* in singular.

```
:LexicalSense_letters_1
  rdf:type ontolex:LexicalSense ;
  rdfs:comment "\"letters\" as \"literary culture\"" ;
  ontolex:usage :Form_letters ;
.
```

An alternative approach could be to allow a sense to be (only) expressed by an instance of the class `Form` that denotes a grammatical number of the associated headword. To this end, the current state of the OntoLex-Lemon model would need to be extended in order to allow a relation (or a property) between an instance of the class `ontolex:Forms` and instances of the class `ontolex:LexicalSense`. This would allow for the direct representation of morpho-semantic variants of the type discussed in this paper. But this second approach would signify a departure from the core module of OntoLex-Lemon, which stipulates that only a lexical entry can be linked to a sense, a concept or an ontological reference.

In both cases, we are able to model together both the information obtained from WordNet and insights derived from the word embeddings. This could lead to a mapping of word embeddings to a computational lexicon. This mapping could be utilized to validate WordNet entries and dynamically create new ones.

6 Discussion

In terms of the creation method for the dataset, we opted for the utilization of simple grammatical rules of inflectional morphology in form of an existing Python package. While this method could be improved on several levels – as for instance utilizing several resources as sources of information, applying more complex morphological components, or analyzing a larger variety of morphosemantic variants – it still returned a significant number of examples

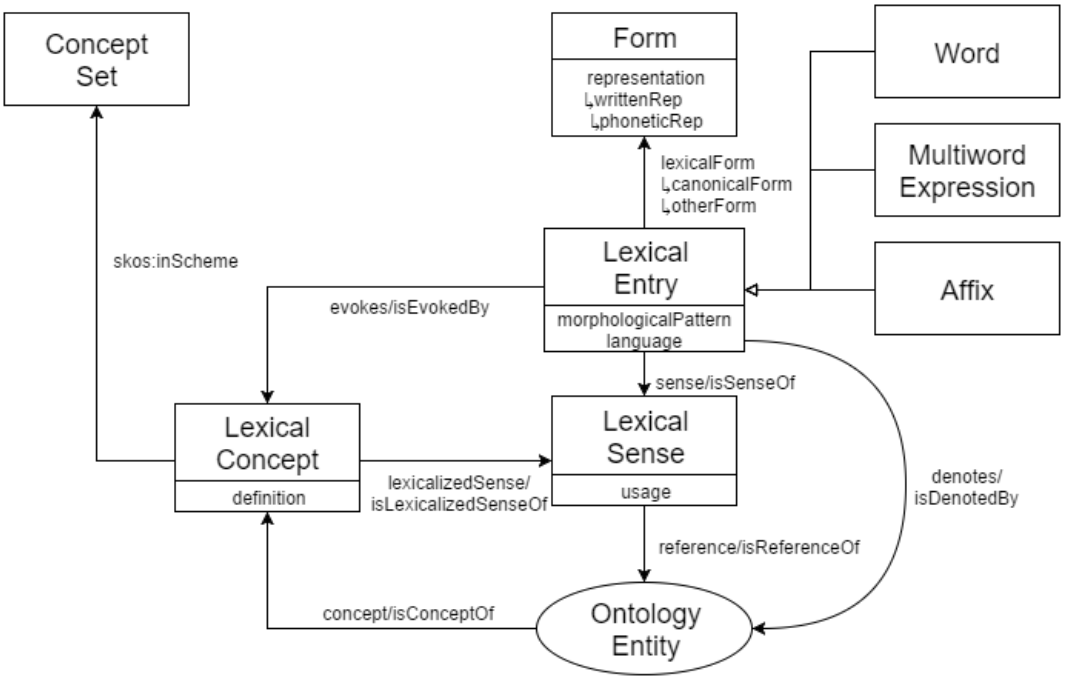


Figure 1 The core module of OntoLex-Lemon: Ontology Lexicon Interface. Graphic taken from `https://www.w3.org/2016/05/ontolex/`.

for the phenomenon under investigation. One central issue of the dataset is the duplication of entries due to more than one plural version of a word, which for now occurred only once with "dominos" and "dominoes" as valid plural versions, but could be aggravated with a larger and more complex dataset. This is one more argument in favor of a more complex morphological analysis tool in the dataset creation process.

For the dataset creation method, WordNet is a very useful resource to identify regular inflectional plurals with additional senses. Applying similar techniques with extended rule sets to other lexical and terminological resources promises to result in a larger and more heterogeneous dataset. For now Wiktionary was utilized to check the separation of plurals that share a meaning with singular and those that have no sense in common with the singular and compare this separation to plurals in the ten nearest neighbors in vector space of two word embeddings. One issue that has to be mentioned here is that modeling of plural-singular connections turns out to be inconsistent in Wiktionary. We consulted plural pages only and categorized plural words into "plural only" senses if no reference "plural of" could be found on the page. However, at times, the reference is missing from the plural page, e.g. "graphics (uncountable)", but a reference to the plural can still be found on the singular page, e.g. "graphic (plural graphics)". Such inconsistent modeling complicate any automated information extraction process.

Thus, in the long run, this separation of plural only and shared meanings of our English noun pairs should be improved upon. One option is the costly manual annotation that might suffer from the complexity of the task, since users may not be familiar with all senses of a word. On the other hand, a corpus-based statistic on the frequency of different plural meanings might provide a more principled analysis of which words are predominantly used in their plural only meaning rather than as a plural of the singular with a shared meaning. In this regard, it would also be interesting to see how to automatically integrate the semantic shift as well as corpus- and vector-based information into OntoLex-Lemon.

In terms of exploring vector space, it would be interesting to repeat our experiments with embedding repositories other than word2vec and fastText. However, to some extend the choice here is limited, since more powerful recent embedding libraries, such as the Bidirectional Encoder Representations from Transformers (BERT) [7], are directed towards words in context and/or sentential embeddings. Querying such contextualized word embeddings with individual words devoid of any context somewhat defeats their purpose.

7 Related Work

While we have presented some studies on inflectional morphology in Section 2.1, this section focuses on computational morphological approaches using embeddings. Analyses of the relation between semantics and morphology have particularly lately been done based on word embeddings. Extensive analogy-based evaluations of morphological and semantic relations in word embedding models across more than 40 categories have shown that inflectional relations are among the best performing ones [9]. Nevertheless, none of them reached an 80% accuracy mark. On the one hand, this could be attributed to the nature of the analogy task and it has been attempted to better adapt the nature of the task to morphological variations [11]. On the other hand, embeddings can be improved by making them morphologically aware, that is, learn embeddings for morphological components (e.g. lemmas, affixes), morphological categories, and word surface forms.

Recent approaches have focused on composing morphologically aware embeddings to improve on the semantic performance of embedding models. Avrahaman and Goldberg [1] adapt fastText [4] to train embeddings for all possible combinations of surface form, lemma, and morphological tags of a word and test on common and rare words. They attribute semantic information to the lemma and morphological information to the affix. In conclusion, they explicitly recommend using lemmas only as for most tasks morphological affixes an be dropped. Nevertheless, their analyses of common words reveals a drop if excluding surface forms (limiting vectors to lemmas and morphological tags), which they attribute to semantic shifts in morphological templates without further investigating the phenomenon, which is exactly where we take over in this paper.

In general, it has been shown that complex composition models tend to outperform simple vector addition or composition methods. Malouf [12] propose a Recurrent Neural Network (RNN) model that predicts complex inflectional classes, which takes a lexeme, set of morpho-syntactic features, and a partial word form as input and outputs a probability distribution for the next segment in the word form in seven morphologically complex languages. Cotterell and Schütze [6] find that approximating a vector with a trained Recurrent Neural Network (RNN)-based model outperforms additive vector composition. However, this approach focuses on derivational morphology, which by definition investigates semantic shifts induced by morphological changes to a word.

In derivational morphology, several linguistic factors have been analyzed in connection with word embeddings. Pado et al. [16] analyze linguistic factors in the ability of Compositional Distributional Semantic Models (CDSMs) to predict distributional vectors for derived word forms given the vectors for their base forms, which they test on 74 derivation patterns in German. Most difficult derivational patterns to predict were found to be those modifying argument structure, semantic irregularities, and within-POS derivation. We believe that more studies in this directions might be in order for the semantic behaviour of inflectional morphology, since those causing semantic shift currently have not been considered in modern approaches.

8 Conclusion and Future Work

We present ongoing work on detecting and formally representing inflectional morpho-semantic variants. While their impact on morphological embeddings has been noted, to the best of our knowledge no comprehensive study has been provided. Our contributions are a dataset of English nominal inflectional morpho-semantic variants of grammatical number and an analysis of their representation in vector space models. One major outcome of this work is the realization that the problem of semantic shifts in inflectional variants of regular morphemes is a significant phenomenon and that it seems that inflectional semantics can be as variable as derivational semantics.

As a second major contribution of this work, we propose a method for representing such variants in a machine readable and formal model called OntoLex-Lemon. To this end, the current version of the model needs to be slightly adapted to account for morpho-semantic variants of grammatical number. This extension can serve as a basis for its potential use for latest neural network based approaches on morphological modeling, such as the potential to include morphological information in training knowledge graph embeddings. Testing these potentials is future work.

For the time being, our approach focuses on English nouns and grammatical number. However, we have good reason to believe that discussed phenomena can be observed in different inflectional morpheme types as well as natural languages, both of which are left as future directions. We also intend to extend our study o different pre-trained embeddings.

References

1 Oded Avraham and Yoav Goldberg. The Interplay of Semantics and Morphology in Word Embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 422–426. Association for Computational Linguistics, 2017.

2 Jiang Bian, Bin Gao, and Tie-Yan Liu. Knowledge-Powered Deep Learning for Word Embedding. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 132–148. Springer, 2014.

3 Steven Bird. NLTK: The Natural Language Toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. Association for Computational Linguistics, 2002.

4 Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

5 Greville G Corbett. Canonical Typolgy, Suppletion, and Possible Words. *Language*, pages 8–42, 2007.

6 Ryan Cotterell and Hinrich Schütze. Joint Semantic Synthesis and Morphological Analysis of the Derived Word. *Transactions of the Association of Computational Linguistics*, 6:33–48, 2018.

7 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805, 2018. `arXiv:1810.04805`.

8 Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, May 1998.

9 Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. Analogy-based Detection of Morphological and Semantic Relations with Word Embeddings: What Works and What doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, 2016.

10 Paul Kiparsky and Judith Tonhauser. Semantics of Inflection. *Handbook of Semantics*, 3:2070–2097, 2012.

11 Tal Linzen. Issues in Evaluating Semantic Spaces Using Word Analogies. In *In Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 13–18, 2016.

12 Robert Malouf. Abstractive Morphological Learning with a Recurrent Neural Network. *Morphology*, 27(4):431–458, 2017.

13 John McCrae, Guadalupe Aguado-de Cea, Paul Buitelaar, Philipp Cimiano, Thierry Declerck, Asuncion Gomez-Perez, Jorge Garcia, Laura Hollink, Elena Montiel-Ponsoda, Dennis Spohr, and Tobias Wunner. Interchanging Lexical Resources on the Semantic Web. *Language Resources and Evaluation*, 46(4):701–719, 2012.

14 John P. McCrae, Paul Buitelaar, and Philipp Cimiano. The OntoLex-Lemon Model: Development and Applications. In Iztok Kosem, Jelena Kallas, Carole Tiberius, Simon Krek, Miloš Jakubíček, and Vít Baisa, editors, *Proceedings of eLex 2017*, pages 587–597. INT, Trojína and Lexical Computing, Lexical Computing CZ s.r.o., September 2017.

15 Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013. `arXiv:1301.3781`.

16 Sebastian Padó, Aurélie Herbelot, Max Kisselew, and Jan Šnajder. Predictability of Distributional Semantics in Derivational Word Formation. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 1285–1296, 2016.

Opening Digitized Newspapers Corpora: Europeana's Full-Text Data Interoperability Case

Nuno Freire
INESC-ID, Lisbon, Portugal
nuno.freire@tecnico.ulisboa.pt

Antoine Isaac
Europeana Foundation, The Hague, The Netherlands
Vrije Universiteit Amsterdam, The Netherlands
antoine.isaac@europeana.eu

Twan Goosen
CLARIN ERIC, Utrecht, The Netherlands
twan@clarin.eu

Daan Broeder
KNAW Humanities Cluster, Amsterdam, The Netherlands
daan.broeder@di.huc.knaw.nl

Hugo Manguinhas
Europeana Foundation, The Hague, The Netherlands
hugo.manguinhas@europeana.eu

Valentine Charles
Europeana Foundation, The Hague, The Netherlands
valentine.charles@europeana.eu

Abstract

Cultural heritage institutions hold collections of printed newspapers that are valuable resources for the study of history, linguistics and other Digital Humanities scientific domains. Effective retrieval of newspapers content based on metadata only is a task nearly impossible, making the retrieval based on (digitized) full-text particularly relevant. Europeana, Europe's Digital Library, is in the position to provide access to large newspapers collections with full-text resources. Full-text corpora are also relevant for Europeana's objective of promoting the usage of cultural heritage resources for use within research infrastructures. We have derived requirements for aggregating and publishing Europeana's newspapers full-text corpus in an interoperable way, based on investigations into the specific characteristics of cultural data, the needs of two research infrastructures (CLARIN and EUDAT) and the practices being promoted in the International Image Interoperability Framework (IIIF) community. We have then defined a "full-text profile" for the Europeana Data Model, which is being applied to Europeana's newspaper corpus.

2012 ACM Subject Classification Applied computing → Annotation; Applied computing → Document metadata; Applied computing → Digital libraries and archives

Keywords and phrases Metadata, Full-text, Interoperability, Data aggregation, Cultural Heritage, Research Infrastructures

Digital Object Identifier 10.4230/OASIcs.LDK.2019.22

Funding *Nuno Freire*: This work was partly supported by Portuguese national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2019, and by the European Commission under contract number 30-CE-0885387/00-80.e.

© Nuno Freire, Antoine Isaac, Twan Goosen, Daan Broeder, Hugo Manguinhas, and Valentine Charles;
licensed under Creative Commons License CC-BY

2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 22; pp. 22:1–22:14

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Cultural Heritage Institutions (CHI), such as libraries and archives, hold collections of printed newspapers of the past centuries. These are valuable resources for historians, linguists and other researchers working in Digital Humanities. The retrieval of printed newspapers' content based on metadata only is a task nearly impossible, however. Cultural Heritage Institutions usually describe the series of a newspaper publication (typically known as "title level" description) and its individual publications ("issues") in their catalogs, but no description of individual articles. The typical use of the catalogs of newspapers is thus only to retrieve issues by date of publication, as there is no detail for effective retrieval of the content at finer-grained levels.

The wide interest in newspapers and the challenges they pose for retrieval has motivated CHIs to prioritize the digitization of their newspapers collections. CHIs also realized that the retrieval of newspapers' content based on machine readable full-text is particularly important, given the unavailability of article level descriptions in the catalogs. Accordingly, CHIs have also sought to apply Optical Character Recognition (OCR) during the digitization process.

Our work addresses the general problem of the retrieval of newspapers in the context of aggregations of digital Cultural Heritage (CH) resources, in particular that of Europeana. Europeana seeks to facilitate the use of resources from and about Europe. It enables access to objects via its Collections portal,[1] which supports all official languages of Europe, and its open APIs enable third-party applications. Europeana is based on metatada provided by its CHI partners and presently holds metadata from over 3,700 CHIs.[2] Providing access to newspapers is relevant to Europeana's mission, especially for promoting the re-use of CH resources for research. Europeana indeed also aims to facilitate research, especially for the digital humanities, via its Europeana Research initiative.[3] This initiative seeks to address issues related to, e.g., licensing, which affect the research re-use of CH metadata and content. In particular, it has identified research re-use of newspapers resources as a key use case, as well as an area with strong system and data interoperability challenges.

Digitized newspapers are Europeana's first case of aggregation and distribution of full-text CH resources. Europeana's systems have relied so far on metadata and links to digitized resources at partners' sites. The Europeana Data Model (EDM) [7] allows it to perform scalable aggregation of (and access to) references to digital representations of CH artifacts with rich context metadata. EDM follows the Linked Open Data principles [1]. An important aspect of EDM is its flexibility and genericity: it can be easily mapped to other (CH) data models and extended [3]. This makes it a potential base for the interoperability of full-text resources within the Europeana ecosystem.

This paper presents how we have tested this assumption by trying to extend EDM to cater for interoperability of full-text CH corpora. The first aim of our work is to support a centralized search engine and rich user interfaces. But we have also investigated the issue of interoperability of full-text between Europeana and research infrastructures (EUDAT and CLARIN). Our work focuses on Europeana and research use, but we claim it has impact on other application contexts, as we sought to align with the generic International Image

[1] https://europeana.eu
[2] https://pro.europeana.eu/files/Europeana_Professional/Projects/Project_list/Europeana_DSI/Deliverables/europeana-dsi-d1.2-amount-of-data-partners-and-outreach-to-major-institutions.pdf
[3] https://research.europeana.eu

Interoperability Framework (IIIF).[4] IIIF is a family of specifications that were conceived to facilitate systematic reuse of image resources in digital repositories maintained by CH institutions. It specifies several HTTP based web services covering access to images, the presentation and structure of complex digital objects composed of one or more images, and searching within their content. IIIF's strength resides in the presentation possibilities it provides for end-users. We present related work on digitized newspapers and the use of CH data in research infrastructures in Section 2. Section 3 presents the exploratory work conducted by Europeana, EUDAT and CLARIN, and the interoperability requirements derived from it. Section 4 presents our EDM extension for full-text, and Section 5 concludes.

2 Related work

Several initiatives exist worldwide with similar target user groups and use cases as Europeana, with respect to aggregation of digitized newspapers. The organizational structure and technical interoperability context of Europeana are quite different, however. For example, Chronicling America,[5] a national aggregation of newspapers in the United States of America, gathers its corpus from the digitization conducted under the National Newspaper Digitization Program. The direct relation of Chronicling America with the digitization process, results in more homogeneous metadata and full-text content to provide access to.

Europeana Newspapers [6] was an earlier project from the Europeana community, which aggregated metadata and full-text content in a portal that, while currently hosted by Europeana, sits on a completely disconnected platform. The project established interoperability by defining a METS/ALTO profile [11], but its application was restricted to the project and did not spread to other CHIs afterwards.

The IIIF Community has conducted similar work to ours in establishing a generic representation of full-text associated with images for the IIIF Presentation API. We participate in a IIIF Newspapers Community Group that gathers IIIF community members working with digitized newspapers. The IIIF representation patterns strongly inspired our work. These, however, are quite generic and the connection with (descriptive) metadata is rather loose in the IIIF presentation API, which relies on linking to document using models like EDM for representing fully-fledged metadata. Furthermore, directly relying on IIIF APIs is an obstacle for the metadata providers who cannot deploy IIIF services for their content.

Regarding interoperability with research infrastructures, related work in CH digitized resources and OCR full-text includes Herbadrop [5]. This initiative works with resources from museums and botanical gardens, which own collections of plant samples with detailed annotations from botanists. Herbadrop has worked with the EUDAT CDI;[6] as part of a data pilot [5].

Finally, some CHIs provide data to CLARIN,[7] in particular university libraries. CLARIN aggregates CH resources in a similar process to Europeana's but uses a different metadata format [4]. Regarding full-text corpora within CLARIN, we observe a prevalence of the Text Encoding Initiative (TEI) format[8] next to plain text content in terms of support by existing tools and also in published research. TEI usage within the Europeana Network is limited: it is only present in CHIs that focus on supporting researchers. Plain text content is often not provided by CHIs.

[4] https//iiif.io
[5] https://chroniclingamerica.loc.gov/
[6] EUDAT Collaborative Data Infrastructure; https://www.eudat.eu/eudatcdi
[7] Common Language Resources and Technology Infrastructure; https://www.clarin.eu/
[8] TEI – Text Encoding Initiative; https://www.tei-c.org/

3 Needs for interoperability with Research Infrastructures

Europeana is interested in investigating how research data infrastructures can facilitate the research use of CH resources. By leveraging on research infrastructures that operate at a European level and across scientific disciplines, it hopes to reach researchers from all scientific disciplines, without having to work with many national and domain-specific research infrastructures or providing its own. We describe here the efforts on the Europeana Newspapers corpus conducted with two infrastructures: CLARIN and EUDAT. This corpus was aggregated from 11 CHIs during the Europeana Newspapers project. It contains metadata descriptions, digitized images and full-text of 911 newspaper titles that, in total, comprise over 11 million pages [6], in multiple languages and scripts. We present, in this section, the interoperability challenges identified and what we did to tackle them.

3.1 Interoperability with CLARIN

CLARIN is a federation of language data repositories, service centers and centers of expertise. CLARIN aggregates metadata and makes the underlying resources discoverable and usable within research workflows. It allows researchers to carry out natural language processing tasks by invoking processing tools directly from its generic user interface. Establishing good interoperability between Europeana and CLARIN can help fitting a large number of CH resources into CLARIN's supported workflows. It will open up new applications for CLARIN's processing tools and promote research incorporating CH resources.

CLARIN carried out a first analysis of the Europeana Newspapers corpus in 2015, establishing goals and a ground for connecting the two infrastructures and full-text interoperability. Later, we sought to address the interoperability issue for metadata [9]. The two infrastructures use specific metadata models: EDM for Europeana and the Component MetaData Infrastructure (CMDI) for CLARIN [4]. Interoperability is achieved via CLARIN's metadata conversion mechanisms, based on a CMDI profile for EDM.[9] Europeana's metadata for Newspapers and other datasets can thus be made available within the CLARIN systems.

The desirable level of interoperability between the two infrastructures has not been achieved, however. The newspapers full-text corpus, although partially discoverable within CLARIN, cannot yet be processed by CLARIN's tools in research workflows. The following requirements for how metadata and full-text content are made available by CHIs were noted and greatly influenced our work on extending EDM for exchanging full-text content:

- Direct links to content files – when CHIs only expose links to websites or viewers in the metadata aggregated by Europeana, the files cannot be processed by CLARIN (and others).
- Technical metadata – information like media type and file size are essential for automated processing workflows and highly desirable for discovery
- Language of the content – most natural language processing tools are language dependent, making the language information carried in CH metadata essential.

3.2 Interoperability with EUDAT

EUDAT is a European infrastructure of integrated data services devoted to scientific and research data storage and life cycle management. It has been developed in close collaboration with over 50 research communities spanning across many different scientific disciplines

[9] Available in CLARIN's component registry: `https://catalog.clarin.eu/ds/ComponentRegistry/#/?itemId=clarin.eu%3Acr1%3Ap_1475136016208`

such as Life Sciences, Humanities, Earth Sciences and Physics, with more than 20 major European research organizations, data centres and computing centres involved. Many of these collaborations are carried out as data pilots providing test-beds that vary in disciplines, communities, project group sizes and technological maturity. Europeana conducted a data pilot with EUDAT that consisted in a case study on the Europeana Newspapers corpus [5]. The general goal was to investigate how EUDAT data services can facilitate the use of CH resources for research purposes. The questions laid out at the start of the data pilot were:

- How can the resources be discovered?
- How can the resources be shared in practical ways for researchers?
- How can advanced computation be applied to these CH datasets?
- How can the resources and datasets be cited and referenced in research?
- How can the CH institutions re-use the outcomes of research?

An evaluation of the available EUDAT services was conducted, using the newspapers corpus as case study. The two infrastructures were successfully interconnected and EUDAT fulfilled the expectations for making the corpus available to researchers and for computational processing. The persistent identification of EUDAT resources also met the citability requirement. The EUDAT service did not scale to the dimension of the corpus, but only due to an underestimation of the required computational capacity during the pilot [5]. Beyond the full-text corpus case study, interoperability was also trialled for metadata-based discovery of CH datasets. Both infrastructures have common underlying technologies that facilitate interoperability, including on modelling full-text, since EUDAT is developing its semantic annotation service based on the W3C Web Annotation Data Model,[10] which is a key component of the EDM extension we are going to present in the next section.

4 Building a full-text profile for the Europeana Data Model

A profile for representing full-text in EDM is a key requirement for achieving a sustainable interoperability framework for full-text CH corpora in Europeana. It has potential applications in full-text aggregation, indexing, user experience and data re-use. This section presents the context, requirements and the EDM full-text profile.

4.1 Context and requirements for designing the data model

Based on the corpus of full-text newspapers, the case studies with research infrastructures and recommendations from the earlier Europeana projects [6, 2], we have identified these requirements:

- The availability of full-text must be stated explicitly in the metadata.
- The representation of full-text should be compatible with the representation of the newspapers' structure (issue, page, article, etc.) in the descriptive metadata.
- The representation of full-text must allow the specification of the language and script of the text, and it should allow this specification to be done at several levels of granularity of the text (e.g. for a paragraph, for a word, etc.).
- URLs to views of the digital objects must be explicitly stated in the metadata.
- Multiple full-text resources must be referenced via direct URLs.
- Resources requiring a protocol to be served need to be clearly identifiable.

[10] https://www.w3.org/TR/annotation-model/

- When more than one full-text resource is associated with a digital object, it should be possible to represent their part-whole relationship.
- When more than one full-text resource is associated with a digital object, it should be possible to represent their sequential order.
- When a full-text resource is available as a fragment of text, the URI or the literal identifying the specific text fragment may be provided in the data.
- When a full-text fragment is available, the image area it refers to should be identified (via coordinates).

The IIIF community has suggested to publish textual representations of (part of) images, such as transcriptions, using annotations from the W3C Web Annotation model (WA). Annotations are included in the IIIF "manifests"[11] of the newspapers, as a list of annotations, each one referring to a portion of the full-text and indicating its corresponding position in the image of a page. Representing full-text as annotations seems the best solution as it can support simple scenarios such as the positioning of a text fragment on an image as well as more complex ones like OCR correction.

This approach, besides its community traction, is compatible with the Linked Data vision and fits well Europeana's use of annotations for other purposes [10]. One of the cases that has recently emerged in Europeana is indeed the representation of manual transcriptions of content.[12] As meeting the requirements of these related cases in similar ways is extremely desirable, we decided to follow the IIIF Community approach. Our modelling exercise thus becomes one of fitting into EDM a representation of the full-text content of newspapers as annotations on the images of newspapers' pages.

4.2 EDM extension addressing the initial full-text requirements

Our extension of EDM for representing full-text follows the recommendations of IIIF (in its coming version 3) and WA. Full-text is represented as the body of an annotation that has as target an image, as illustrated in Figure 1. We model the image as an `edm:WebResource` (the usual EDM approach) and the text itself as a new proposed subclass of `edm:WebResource`, `edm:FullTextResource`.[13] Figure 2 illustrates the simplest case. Annotations are modeled using WA's `oa:Annotation` class and `oa:hasBody` and `oa:hasTarget` properties. Annotations used for representing full-text must have the property `oa:motivatedBy` with the value `edm:transcribing`, distinguishing them from Europeana annotations used for other motives, as well as following IIIF's latest best practices[14] (NB: we omit it from our figures for readability reasons).

The extension supports two levels of detail for associating the full-text with the image: with and without its position within the image. The text can also be provided by value (a plain literal) or by reference (as a URI, and/or as a selection/extract from another text resource). The following sections present the details of these options.

[11] IIIF manifests are "the overall description of the structure and properties of the digital representation of an object."; `http://iiif.io/api/presentation/2.0/#primary-resource-types`

[12] Cf. Europeana's initiative on transcribing WWI-related content; `https://transcribathon.com/`

[13] The full-text comes as `rdf:value` for the `edm:FullTextResource`, using WA's "embedded text" pattern (`https://www.w3.org/TR/annotation-model/#embedded-textual-body`) with a type independent from the resource's being used in an annotation, unlike WA's `oa:TextualBody`.

[14] Cf. IIIF API issue 1258: `https://github.com/IIIF/api/issues/1258`

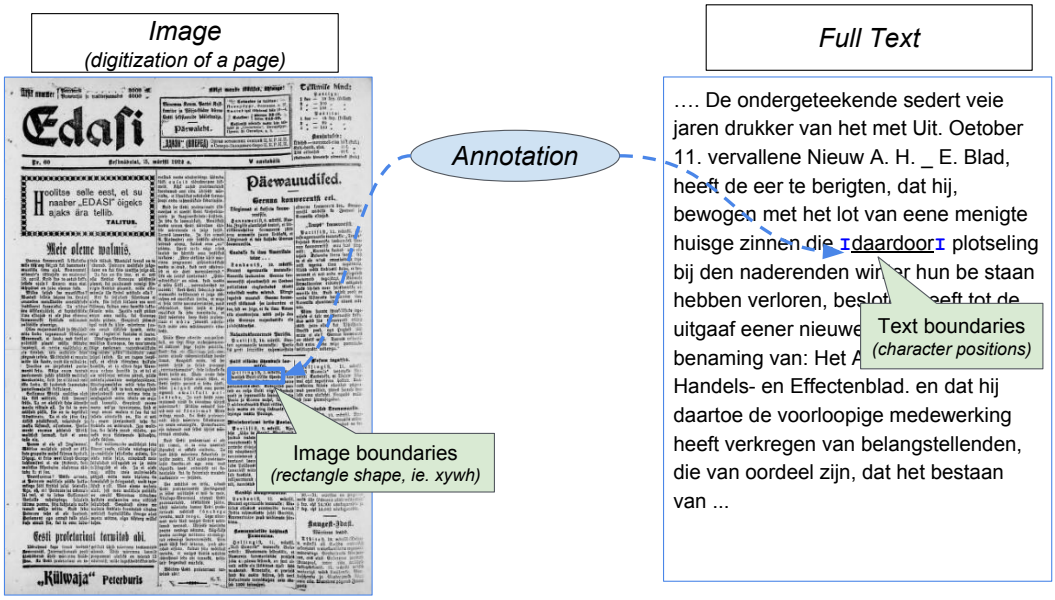


Figure 1 General principles for full-text annotations in the EDM extension.

4.2.1 Full-text without position

In the simplest case, illustrated in Figure 2, full-text is associated with an image without any information about the position of the text within the image.

4.2.2 Full-text associated with fragments with a position in the image

An earlier analysis of newspapers corpora [2] has shown that full-text is sometimes represented as several fragments of text, each referring to a specific area of an image (an article, a specific line in the text or a word). In this case, the full-text fragment is accompanied with coordinates indicating its position on the image.

To support this requirement, we introduce in the model the `oa:SpecificResource` that "is used in between the Annotation and the body or target, as appropriate, to capture the additional description of how it is used in the annotation" [9]. An `oa:FragmentSelector` is applied as selector within the `oa:SpecificResource` to restrict the original target (the `edm:WebResource`) to the specific area to which the text, or fragment, corresponds. Figures 3, 4 and 5 show examples of this solution.

In Figure 3, the `edm:FullTextResource` consists of a fully-fledged resource that corresponds to a paragraph whose position is indicated by the `oa:FragmentSelector`. Note that for rectangle areas, coordinates in the `oa:FragmentSelector` must follow the Media Fragments W3C recommendation and be the subject of a `dcterms:conformsTo` statement referring to `http://www.w3.org/TR/media-frags/` (not shown in the figure).

4.2.3 Full-text selections represented as fragments with a position in the image

Figure 4 shows how more details – in this case, the position of a particular word – can be specified for the association between full-text and images. The area is indicated using the pattern already seen in Figure 3, but the paragraph fragment that corresponds to the

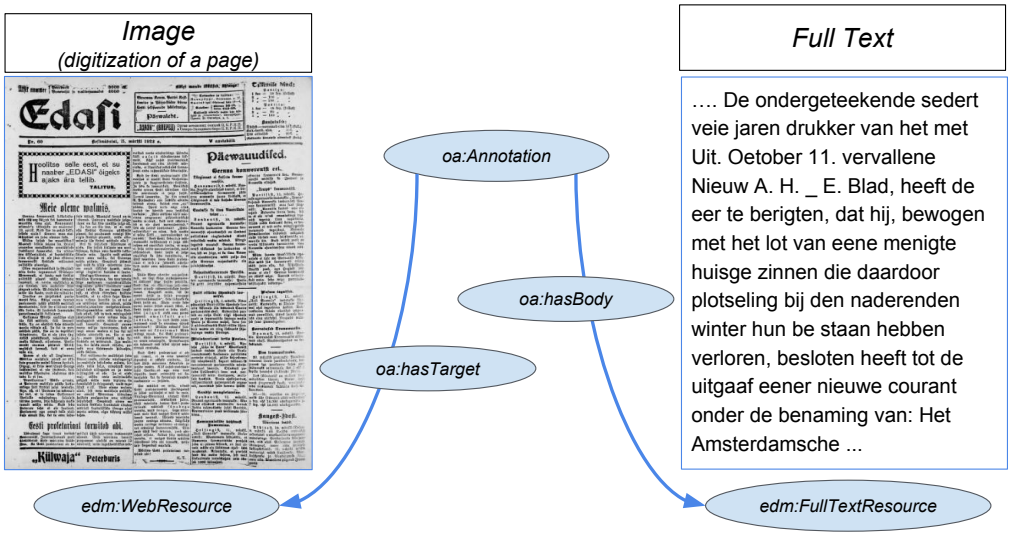


Figure 2 Full-text without position information.

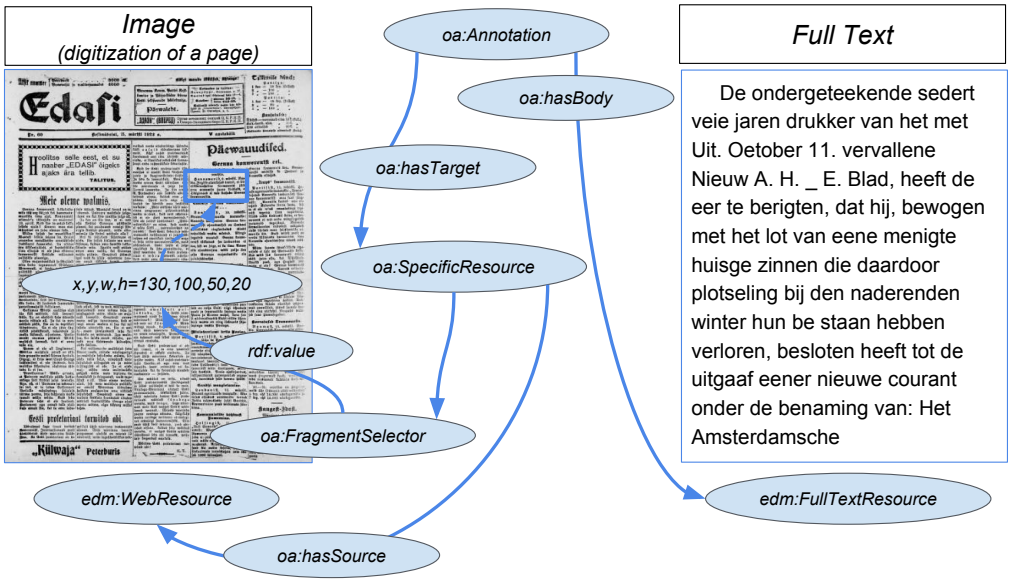


Figure 3 Full-text resource with position on the image.

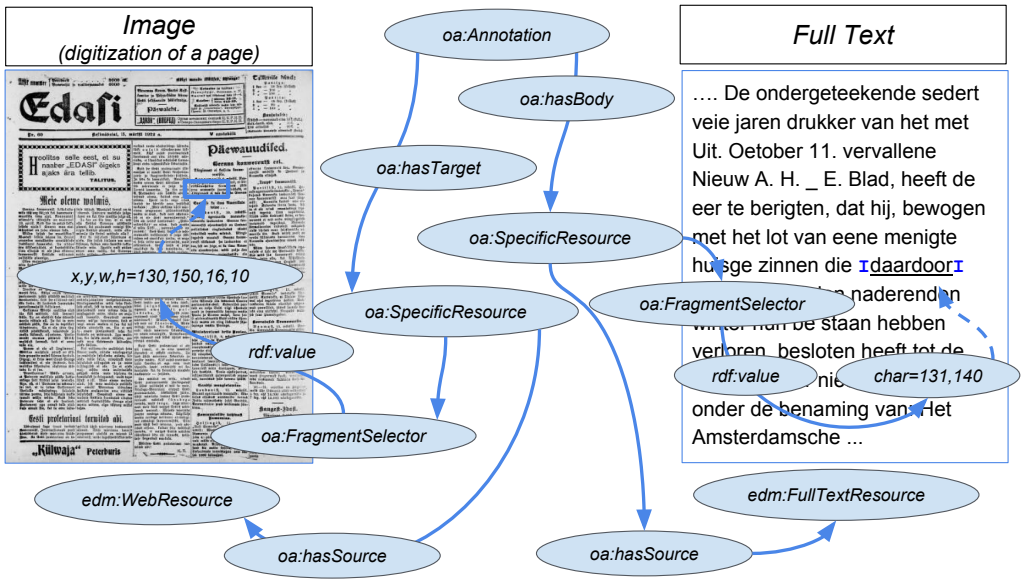


Figure 4 Full-text fragment with position on the image using *oa:FragmentSelector*.

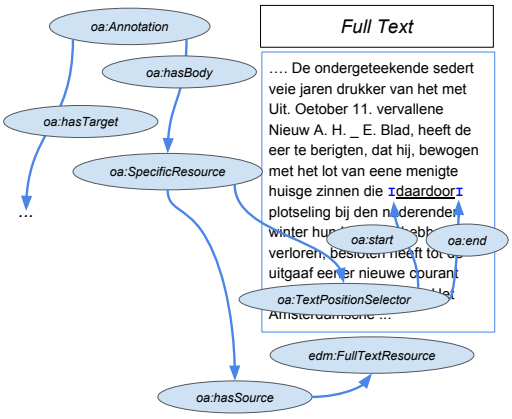


Figure 5 Full-text fragment with position using *oa:TextPositionSelector*.

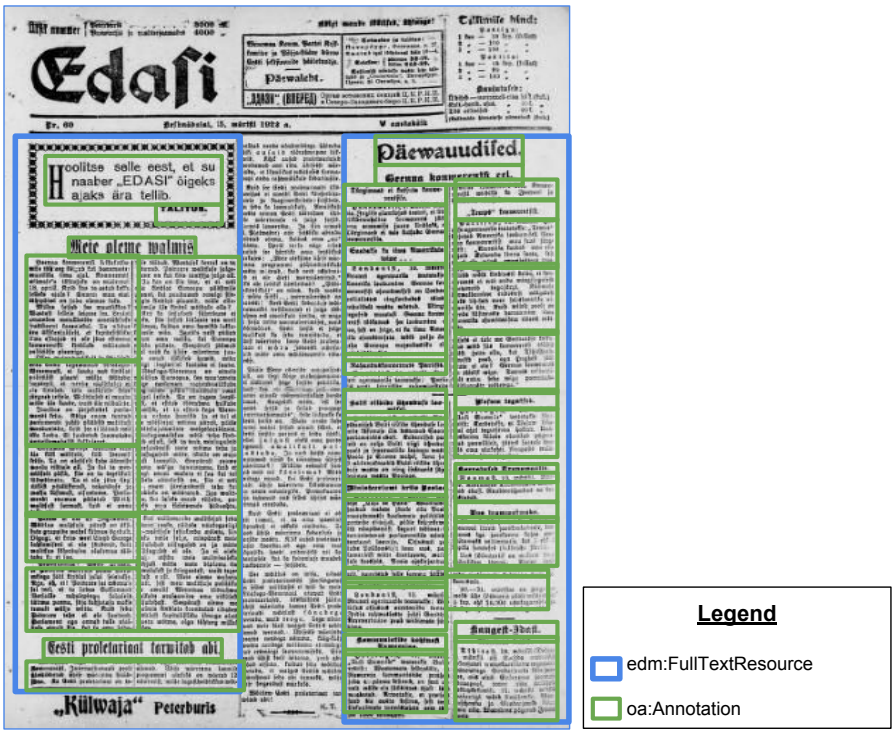


Figure 6 Representing the logical structure of articles and paragraphs of full-text with *edm:FullTextResource* and *oa:Annotation*.

word in the full-text is also given: an `oa:SpecificResource` is created to represent how the textual body of the annotation is derived from another resource. An `oa:FragmentSelector` resource describes the range of text by recording the first and last characters' positions within the source. The `oa:FragmentSelector` must follow RFC 5147 and be the subject of a `dcterms:conformsTo` statement referring to `http://tools.ietf.org/rfc/rfc5147` (not shown in Figure 4). Note that the WA model offers alternatives for representing fragments: e.g., for text fragments, the data from Figure 4 can also be represented using an `oa:TextPositionSelector`, recording the start and end positions with specific properties (see Figure 5). We have decided for now to be flexible in what Europeana will accept, opening the possibility to use equivalent WA selectors. But we will seek to normalize the data we publish, i.e. retaining only one of the options – yet to be discussed with the community.

4.2.4 Logical structure of the full-text

Some digitization efforts apply segmentation techniques to detect the independent sections (such as articles) within a newspaper page. Our EDM extension allows representing the different sections in the full-text. First, text of different levels can be represented as different `edm:FullTextResources` connected across levels using Dublin Core `dcterms:hasPart` and `dcterms:isPartOf` properties. EDM allows this for any digital representation, and this pattern can be used in particular between a newspaper file that contains several pages (images) and the image of each page. In this case, however, text is duplicated across levels. An alternative is to represent the logical structure via the organization of `edm:FullTextResources` and `oa:Annotations`. Our extension assumes that each `edm:FullTextResource` can reflect

a section within a page and act as grouping for all related `oa:Annotations`. Figure 6 shows a newspaper page where two `edm:FullTextResources` represent two articles in the page. It also highlights how (targets of) `oa:Annotations` represent the paragraphs within each `edm:FullTextResource`.

4.2.5 Specifying the language of the full-text

The profile allows the indication of language of the full-text at several levels of detail. At the most general level, the language indicated in the data for the original cultural object (using Dublin Core's `dc:language` property on EDM's `edm:ProvidedCHO` resource[15]) can be seen to apply to the whole full-text as well. Our profile assumes that when a (sub-component of) the full-text does not specify its language, then it inherits the language from the higher levels of its hierarchy. This pattern enables to represent cases when a word in one language is present within a text in another language. But there can be different languages, or a data publisher may prefer to express precise information that does not depend on implicit "propagation" rules between levels in the data. Therefore, the language may be specified at the level of any `edm:FullTextResource`, using an RDF language tag on the `rdf:value` of the resource or the `dc:language` property.[16] At the finest level of detail, languages may be specified on the `oa:SpecificResource` referring to text fragments. Figure 7 illustrates using `dc:language` on the `edm:FullTextResource` and the `oa:SpecificResource`.

4.3 Application of the profile

At this time, the EDM full-text profile is already applied at production level. Europeana has converted the Europeana Newspapers corpus to the EDM full-text profile, therefore, the profile has been applied to more than 11 million pages of newspaper full-text transcriptions, in multiple languages and scripts. Since this corpus originates from data providers from different countries using different practices for digitisation, we see this application as evidence that the model can accommodate the different ways of structuring full-text in digitised objects.

Europeana has also made significant steps implementing the full-text profile in its systems. It has adapted its data infrastructure to support the ingestion of full-text according to the profile (no support for full-text existed previously in Europeana).

Regarding indexing and retrieval of full-text EDM data, Europeana has completed a first version of its solution, which combines the joint retrieval of resources described by metadata only, with resources with full-text and metadata.

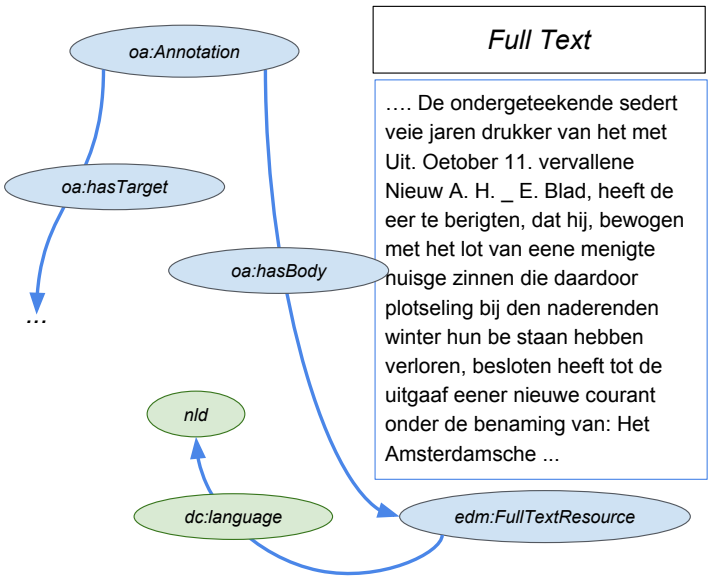
It has completed a first version of indexing and search services, which provides retrieval of full-text resources. This first version is not yet integrated with the main search systems of Europeana (that works only on metadata), but the first steps have taken place for investigating a solution for accomplishing a joint search system.

On top of this, Europeana's final products are a portal and an API. The portal is specialised for the newspapers corpus[17] and provides a user-interface based on full-text retrieval and the association, via image coordinates, between digitised images and the

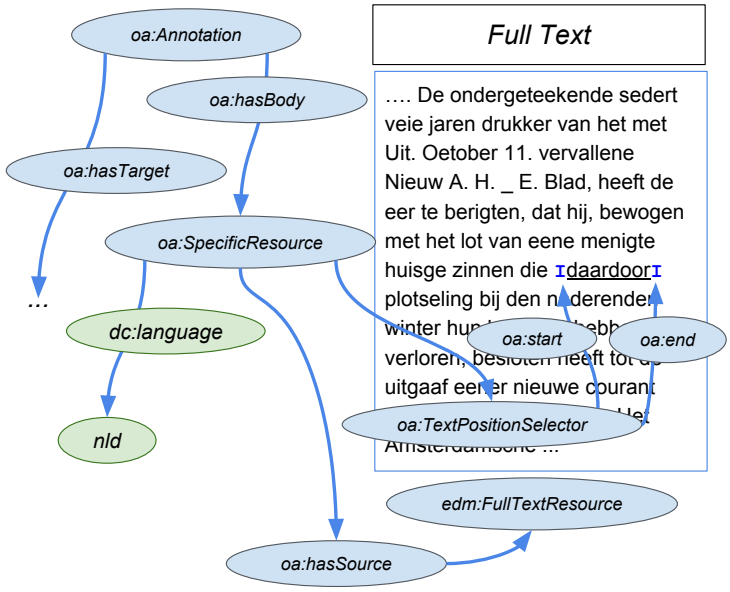
[15] `ProvidedCHO` stands for "Provided Cultural Heritage Object". It is the original object that is described. It may be either a physical object (painting, book, etc.) or digital-born object.

[16] Here again there are two equivalent modeling alternatives: the "traditional" RDF one (already used in EDM and one preferred by the WA model. We intend to accept both and publish both in parallel, but this choice is still open to community feedback.

[17] `https://www.europeana.eu/portal/en/collections/newspapers`



(a) for the whole *edm:FullTextResource*



(b) for a piece of text in isolation (i.e., a word)

Figure 7 Specification of the language of the text.

transcription. This interface uses the full-text to down to word-level detail (when word level coordinates have been recorded during digitisation and OCR). The API service now available for newspapers[18] complements the existing Europeana API with functionality specialised in full-text search and access, including making the full-text available according to the IIIF Presentation API - where the IIIF output is generated from the EDM representation. This improves Europeana's capacity to promote data re-use of CH content through research infrastructures and other target user groups.

5 Future Work and Conclusion

Europeana's investigations in exploring its newspapers full-text corpus with research infrastructures has provided valuable input for making CH corpora better discoverable, accessible, machine processable and citable in research contexts. The requirements identified for research usage of CH full-text corpora support several aspects of the current strategy of Europeana towards improving data quality and direct access to the media contents of CH digital objects [8].

The currently aggregated full-text corpus of Europeana Newspapers has not grown since the end of the Europeana Newspapers project, and an aggregation process based on the ALTO profile was not possible to establish in a sustainable way at Europeana, due to its high technical complexity for adoption by data providers, and also for aggregators. The new model, being based on EDM and following the IIIF Community approach is expected to lower the technical barriers to establish a sustainable full-text aggregation process.

In the near future, our EDM full-text profile is going to be used as the basis to resume the aggregation processes of full-text newspapers content across the Europeana Network. In parallel, we will update the EDM full-text profile, by devising a more precise approach to the modeling alternatives that the current version allows – we have already begun to actively seek feedback from the IIIF Newspapers community. We will also tackle new requirement that could emerge during its adoption: for example, some Europeana stakeholders have voiced interested in an explicit representation of the granularity of the full-text (page, article, paragraph, line, word).

Regarding the re-use of CH full-text data for research, CLARIN is starting an assessment of the applicability of the full-text content, as disseminated by Europeana, to its infrastructure and the connected tools in the context of various typical research use cases, covering resource discovery, retrieval and processing. On basis of the findings of this assessment, we expect to be able to fine-tune the full-text profile and the content APIs on the side of Europeana, and adapt the exploitation of Europeana's services by CLARIN accordingly, so as to achieve a broad integration of large volumes of full-text content with real-world applicability for the social sciences and humanities research communities.

References

1 Timothy Berners-Lee. Linked Data Design Issues. W3C-Internal Document, 2006.
2 Valentine Charles, Nuno Freire, Hugo Manguinhas, Peter Vos, and Glen Robson. Recommendations for enhancing EDM to represent digital content. Technical report, Europeana Cloud D4.4, 2016.
3 Valentine Charles and Antoine Isaac. Enhancing the Europeana Data Model (EDM). Technical report, Europeana V3.0, 2015.

[18] https://pro.europeana.eu/data/newspapers-getting-started

4 CMDI Taskforce. *Component Metadata Infrastructure (CMDI) Component Metadata Specification Version 1.2*, 2016.

5 Pascal Dugenie, Nuno Freire, and Daan Broeder. Building new knowledge from distributed scientific corpus: HERBADROP & EUROPEANA: Two concrete case studies for exploring big archival data. In Jian-Yun Nie, Zoran Obradovic, Toyotaro Suzumura, Rumi Ghosh, Raghunath Nambiar, Chonggang Wang, Hui Zang, Ricardo A. Baeza-Yates, Xiaohua Hu, Jeremy Kepner, Alfredo Cuzzocrea, Jian Tang, and Masashi Toyoda, editors, *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*, pages 2231–2239. IEEE Computer Society, 2017. doi:10.1109/BigData.2017.8258174.

6 Alastair Dunning, Alena Fedesenka, Anastasia Gasia, and Markus Muhr. Report on newspapers data aggregated by The European Library. Technical report, Europeana Newspapers D4.5, 2015.

7 Europeana Foundation. *Definition of the Europeana Data Model v5.2.8*, 2017.

8 Europeana Foundation. *Europeana Publishing Guide v1.5*, 2017.

9 Twan Goosen, Dieter Van Uytvanck, and Nuno Freire. Results and Impact of Sharing Europeana Data with CLARIN. Technical report, Europeana DSI-2 MS2.2, 2017.

10 Sergiu Gordea, Hugo Manguinhas, Antoine Isaac, Valentine Charles, Maarten Brinkerink, Alessio Piccioli, and Breandán Knowlton. Modelling and exchanging annotation for Europeana projects. In *Semantic Web in Libraries Conference 2015*, 2015.

11 Günter Mühlberger. METS ALTO Profile (ENMAP). Technical report, Europeana Newspapers D5.2, 2014.

Automatic Detection of Language and Annotation Model Information in CoNLL Corpora

Frank Abromeit
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
abromeit@em.uni-frankfurt.de

Christian Chiarcos
Applied Computational Linguistics Lab, Goethe University Frankfurt, Germany
http://acoli.informatik.uni-frankfurt.de/
chiarcos@informatik.uni-frankfurt.de

Abstract

We introduce *AnnoHub*, an on-going effort to automatically complement existing language resources with metadata about the languages they cover and the annotation schemes (tagsets) that they apply, to provide a web interface for their curation and evaluation by means of domain experts, and to publish them as a RDF dataset and as part of the (Linguistic) Linked Open Data (LLOD) cloud. In this paper, we focus on tabular formats with tab-separated values (TSV), a de-facto standard for annotated corpora as popularized as part of the CoNLL Shared Tasks. By extension, other formats for which a converter to CoNLL and/or TSV formats does exist, can be processed analoguously. We describe our implementation and its evaluation against a sample of 93 corpora from the Universal Dependencies, v.2.3.

2012 ACM Subject Classification Information systems → Structure and multilingual text search

Keywords and phrases LLOD, CoNLL, OLiA

Digital Object Identifier 10.4230/OASIcs.LDK.2019.23

Category Short Paper

Supplement Material https://annohub.linguistik.de

Funding The research described in this paper was conducted in the context of the Specialized Information Service Linguistics, funded by German Research Foundation (DFG/LIS, 2017-2019). The contributions of the second author were conducted with additional support from the Horizon 2020 Research and Innovation Action "Pret-a-LLOD. Ready-to-use Multilingual Linked Language Data for Knowledge Services across Sectors" (H2020-ICT-2018-2, 2019-2021).

1 Introduction

The *lin|gu|is|tik.de* portal is a virtual library which provides a rich, manually curated bibliography for linguists, coupled with inter-library search, library catalogues, indices of electronic resources, as well as various services supporting research in the language sciences [2].[1] Since 2015, we have been extending this service with respect to indexing and search over language resources, initially for language resources data provided as part of the (Linguistic) Linked Open Data (LLOD) cloud [3],[2] i.e., using RDF as a data format, HTTP URIs for identifying elements of linguistic analysis, and open licenses for data publication.

This functionality is currently being extended for indexing language resources in other popular formats. Such data is available in greater numbers than RDF-native language

[1] https://www.linguistik.de/
[2] http://linguistic-lod.org/


© Frank Abromeit and Christian Chiarcos;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 23; pp. 23:1–23:9

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

resources, however, much of its information is implicit: In particular, RDF features explicit markers for the language of a particular string (language tags), and also URIs to identify grammatical features and linguistic annotations across different data sets, e.g., using vocabularies such as the Ontologies of Linguistic Annotation (OLiA),[3] the General Ontology of Linguistic Description (GOLD),[4] or the lexinfo model for grammatical features in lexical resources.[5] In conventional formats, such information is often missing, and if not provided as part of the formal metadata, it needs to be inferred from the data itself. In this paper, we describe a method for the automatic detection of language and annotation metadata from popular one-word-per-line (OWPL) formats, where rows correspond to individual words, and columns correspond to annotations of a particular type each. Because of their popularity, we specifically focus on CoNLL and related TSV formats as commonly used in corpus linguistics, lexicography and natural language processing.

AnnoHub is a web application that provides services to analyze language resources like corpora in RDF, CoNLL and XML formats with respect to the used annotation schemes and present languages, and to curate and publish such data. The AnnoHub web application is specifically designed to facilitate the workflow of librarians and domain experts involved in creating bibliographical records for language resources and scientific publications, it is thus internally available, only, at the moment. The resulting RDF meta data and the underlying technology stack will be published under an open license with the end of the project. Our implementation builds on – and complements – existing open source software on mapping CoNLL data to RDF [1, CoNLL-RDF].[6]

2 Automated language detection

The language detection was implemented with the *Optimaize* Java library[7] which provides n-gram-based language classification. Natively, it supports the detection of 71 languages. In order to extend the detection to other languages the library provides a tool to build new languages profiles with a text sample from a specific language. We build 444 additional language profiles from a set of about 1.500 machine-readable Bible texts created as part of earlier research.[8]

Given the large number of language models and for reasons of scalability, we perform language detection on a random sample of only 15 sentences from each CoNLL TSV file. As we aim for a generic implementation, and the position of WORD and LEMMA columns varies across different CoNLL dialects, we test *every* column for all languages (and all annotation models, see below). The language profile with the highest probability score for the majority of the 15 sentences was then selected. In some cases, increasing the set of test sentences might improve results, but for reasons of scalability, this was not tested. For detailed results we refer to section 5.

[3] `http://purl.org/olia`
[4] `http://linguistics-ontology.org/`
[5] `https://lexinfo.net/`
[6] `https://github.com/acoli-repo/conll-rdf`
[7] `https://github.com/optimaize/language-detector`
[8] Selected results of this conversion and edition project have been described by Chiarcos et al. [5], although restricted to a subset of Germanic languages. For reasons of copyright, and due to the lack of a fair use principle in German legislation, we were not able to disseminate the data. Instead, we provide build scripts for several major Bible aggregation portals in our Github repository (`https://github.com/acoli-repo/acoli-corpora/tree/master/biblical`), covering about 50% of the internally available data.

2.1 Evaluation

We evaluate our implementation on a sample of 93 corpora from the Universal Dependencies (UD) collection, v.2.3 [6].[9] In the CoNLL-U format that these corpora follow, WORD and LEMMA columns are second, resp. third column, and a summary for the language detection test over these is presented in Tab. 1. Overall, we achieved 84% accuracy (including non-detection of languages for cases where no text was provided, e.g., for ESL data).

Table 1 Result summary for language detection.

	Result	Comment
Match	78/93 (83.88%)	correct language or no language (if not present)
Partial match	3/93 (3.22%)	language correctly recognized for one of the max. 2 text columns
Weak match	2/93 (2.15%)	language found among the top 4 but not top match
Fail	2/93 (2.15%)	language was not correctly identified
No profile	8/93 (8.6%)	no language profile for the language available

Reasons for mis-classification among known languages are the proximity of certain language varieties, e.g., different varieties of Norwegian (nno/nob), the close relationship among historically closely languages (and orthographies) such as Russian (rus) and Bulgarian (bul), or Serbian (srp) and Croatian (hrv), or the relative proximity of different historical stages of the same language in the case of Ancient Greek (grc) and Modern Greek (ell). Another source of errors is that the language models are trained on texts, but that the LEMMA column contains uninflected forms only. Thus, the LEMMA column is more likely to be incorrect than the WORD column. Such errors need attention and should be (and can be) manually corrected by the user. If manual selection among the top matches for a column is allowed (and correctly applied), the accuracy can be increased by 5% to up to 89%, with unrecoverable errors going back to mis-classification (*Fail*, 2.15%) and missing language profiles (8.6%).

3 Automated detection of annotation models

Our approach to detect and disambiguate annotation models builds on the Ontologies of Linguistic Annotation [4, OLiA].[10] The OLiA ontologies provide a formalized, machine-readable view on linguistic annotations for more than 75 different language varieties, they cover morphology, morphosyntax, phrase structure syntax, dependency syntax, aspects of semantics, and recent extensions to discourse, information structure and anaphora, all of these are linked with an overarching reference terminology module. OLiA includes several multi-lingual or cross-linguistically applicable annotation models such as the Universal Dependencies (77 languages), EAGLES (11 European languages), Multext-East (16 Eastern European and Near Eastern languages).

An OLiA annotation model for a given annotation scheme (tagset) provides a formalization in terms of an ontology that defines tags (grammatical features) as instances of ontological concepts. An example for such a definition is given in Fig. 1 for the part-of-speech tag ADJ for an adjective in Morphisto, an annotation model for inflectional morphology in German[7].

[9] `https://universaldependencies.org`, `https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2895`
[10] `http://purl.org/olia`

```
@prefix system: <http://purl.org/olia/system.owl#> .
@prefix : <http://purl.org/olia/morphisto.owl#> .

:ADJ system:hasTagContaining "|ADJ"^^xsd:string ;
     system:hasTagStartingWith "ADJ"^^xsd:string ;
     a :SyntacticAdjective ;
     rdfs:comment "\"proper\" adjectives"^^xsd:string .
```

Figure 1 Definition for the part-of-speech tag ADJ in `http://purl.org/olia/morphisto.owl`.

The definition in Fig. 1 declares `http://purl.org/olia/morphisto.owl#ADJ` as an instance of the class `http://purl.org/olia/morphisto.owl#SyntacticAdjective` and assigns the annotation string "ADJ" to it. An individual does not have to correspond to a particular string (`system:hasTag`), but it can also be defined by a partial match (`system:hasTagContaining`, `system:hasTagStartingWith`, `system:hasTagEndingWith`) or a Perl-style regular expression (`system:hasTagMatching`).[11] For every annotation model, an OLiA linking model defines relationships between classes/properties in the respective annotation model and the OLiA reference model. In that way a connection between the occurrence of a annotation in a corpus and the OLiA reference model which specifies a common terminology that different annotation schemes can refer to can be established. This enables for example a SPARQL search that looks for realizations of adjectives in a RDF resource independently from the annotation scheme used in that resource – as long as an OLiA annotation model for that scheme exists.

3.1 Implementation

In a first step we build a graph database (model graph) from all OLiA annotation models. The *model graph* is a simplified version of the OLiA RDF graphs. It mainly serves 3 purposes:

- Store classes, attributes and relations of all OLiA annotation models
- Store results - annotations in CoNLL resources which could be linked to OLiA
- Enable annotation scheme detection via database queries

The *model graph* contains only 3 types of vertices: CLASS vertices are equivalent to RDF class definitions. TAG vertices contain the string value that is attached to an RDF class/individual (see Fig. 1) via a RDF property like (`hasTag, hasTagStartingWith, hasTagEndingWith or hasTagContaining`). Finally HIT vertices contain the annotation string that is found in a CoNLL file.

The following algorithm describes the steps to determine a best fitting annotation scheme for a given CoNLL column. Before the algorithm can start the set of annotations from that column has to be extracted. A annotation can be a single token (e.g. ADJ) but can also be of the form of a sequence of syntactical or morphological features, e.g. SG-IND-NOM. The input of the algorithm is then the serialization of the individual tokens in such expressions. It should be noted that the upper bound for the input size of the algorithm is the number of different annotations that are found in a CoNLL column.[12] Steps 3 and 8 in the algorithm are query operations on the model graph. Since the TAG vertices are directly connected to CLASS vertices via an edge, the query operation in step 8 is very cheap. The query in step

[11] As an example for a regular expression, consider ^AJ...1.* for `http://purl.org/olia/eagles.owl#NominativeCase`.

[12] Test files commonly contained up to 100 different annotation types per column.

Algorithm 1 Annotation model detection.

1: Extract tokens_i={Annotations from CoNLL file column i}
2: **For t** in tokens_i :
3: **Try** to match **t** with the string/regex of a TAG vertex in the model graph
4: **If** (**t** matches TAG_j) **then**
5: **1.** Insert a new vertex HIT_t into the model graph
6: **2.** Insert an edge from HIT_t to TAG_j
7: **For h** in h_i= {HIT vertices that were created from tokens_i} :
8: Compute z_i = {CLASS vertices that are connected to h via a path in the model graph}
9: **For z** in z_i :
10: **If z** belongs to annotation model X then $\text{count\_AM}_X = \text{count\_AM}_X + 1$
11: **Output** $\max(\text{AM}_X)$

3 is most expensive when an OLiA annotation model defines an annotation in terms of a partial match or (worse) a regular expression. Finally, in step 10 the found CLASS vertices are summed up with respect to the OLiA annotation model they belong to. In Fig. 2, the models SUC, BROWN, MAMBA, GENIA, QTAG and PENN would receive (+1) in step 10 of the algorithm.

```
HIT : DT matches CLASS : http://purl.org/olia/suc.owl#dt
HIT : DT matches CLASS : http://purl.org/olia/brown.owl#DT
HIT : DT matches CLASS : http://purl.org/olia/mamba-syntax.owl#determiner
HIT : DT matches CLASS : http://purl.org/olia/genia.owl#DT
HIT : DT matches ClASS : http://purl.org/olia/qtag.owl#DT
HIT : DT matches CLASS : http://purl.org/olia/penn.owl#DT
```

Figure 2 Different choices to match the tag DT for determiner.

3.2 Evaluation

Table 2 summarizes evaluation results for annotation model detection for four annotations columns (C-4, C-5, C-6, C-8) in 93 UD corpora. Again, the same test data was used. For Universal Dependencies corpora, we focus on parts of speech and dependency labels, i.e., CoNLL-U columns UPOS (UD-style parts of speech, column C-4), XPOS (native parts of speech, column C-5), FEATS (UD dependency labels, column C-6) and DEP (UD dependency labels, column C-8).

Throughout all datasets, UPOS, FEATS and DEP are correctly detected, for evaluating annotation model accuracy, we thus focus on C-5 (XPOS). A challenging aspect is that OLiA does not support all native tagsets for the 77 UD languages. As a measure to estimate the quality of a predicted annotation model for a CoNLL column *c* we introduce the *Coverage* measure which is defined by :

$$\text{Coverage}(\text{model}_X) = \frac{\#\text{ annotations in c found in OLiA annotation model X}}{\#\text{ annotations in c found in any OLiA annotation model}}$$

The Coverage measure ignores annotations that were not recognized in any OLiA annotation model.[13] As tags tend to re-occur in different tagsets, we applied a restrictive filtering to models with a Coverage of more than 95%, so that we achieved a precision of 81.25%.

[13] Note that this allows to identify missing annotations (tags) in OLiA annotation models. Unmatched annotations are displayed in the application user interface.

Table 2 Overview of model column detection (baseline 374 possible model columns).

Criterion/Comment	Columns with a predicted model
Model detected with Coverage > 95%	200/374 (53.4%)
Model detected with Coverage > 80%	314/374 (84%)
Model detected with Coverage ≤ 80%	32/374 (8.5%)
No annotation model was detected	28/374 (7.5%)
A text column was regarded as a model column	3

In cases where no annotation model could be detected, an appropriate OLiA annotation model was missing.[14] Since annotations are highly ambiguous (e.g. same tag can be used in multiple annotation models) other properties of these models need to be incorporated in the detection process. One possibility is to include language information into OLiA annotation models because many annotation models were specifically designed for certain languages. At the moment, such information is not provided by OLiA in order to support adaptations of existing annotation models to linguistically or culturally related language varieties.

4 Editor for manual curation and verification

Aside from the detection routines described above, the AnnoHub infrastructure provides a web front-end that features an editor for the interactive curation and verification of language and annotation model predictions. Its functionality is illustrated here with respect to annotation model detection. For language detection, analoguous views are provided. The model editor can be used to review the computed best fitting models for a CoNLL resource and to correct errors by selecting a different model manually. The editor window (Fig. 3) gives an overview of possible annotation models for a specific column in a CoNLL file. On top of the list the model with the best fitting for all tags in a column is listed. Further results include the following values : a) coverage of occurring tag types in %, b) the number of different found tags, c) the total number of instances for all tags in b), d) the number of tags types that are matched exclusively by this model, e) the total number of instances for all tags in d).

As an example consider the result for column 4 for the CoNLL file en_ewt-ud-train.conllu (Fig. 3). At the top of edit window the PENN annotation model is shown as the selected model for that column. Detailed results for each candidate model can be displayed by expanding a row in the table. In the example the results for the EMILLE annotation model are displayed. In the first column (Found tag/Class) the only part-of-speech tag CC that could be matched in the corpus is listed. In the second column a URL shows the ontology class were a definition for the part-of-speech tag CC can be found and the third column shows the number of found instances for that tag (74). Finally the entry ZERO MATCH displays those annotations that could not be found in the OLiA annotation model for EMILLE together with their count (31) in the last column.

[14] This includes cases where the XPOS column provided POS tags concatenated with other annotations, e.g., for grammatical features. With an OLiA annotation model expecting POS tags to occur in isolation, rule-based preprocessing of XPOS annotations is necessary to produce a match. This has not been attempted, so far.

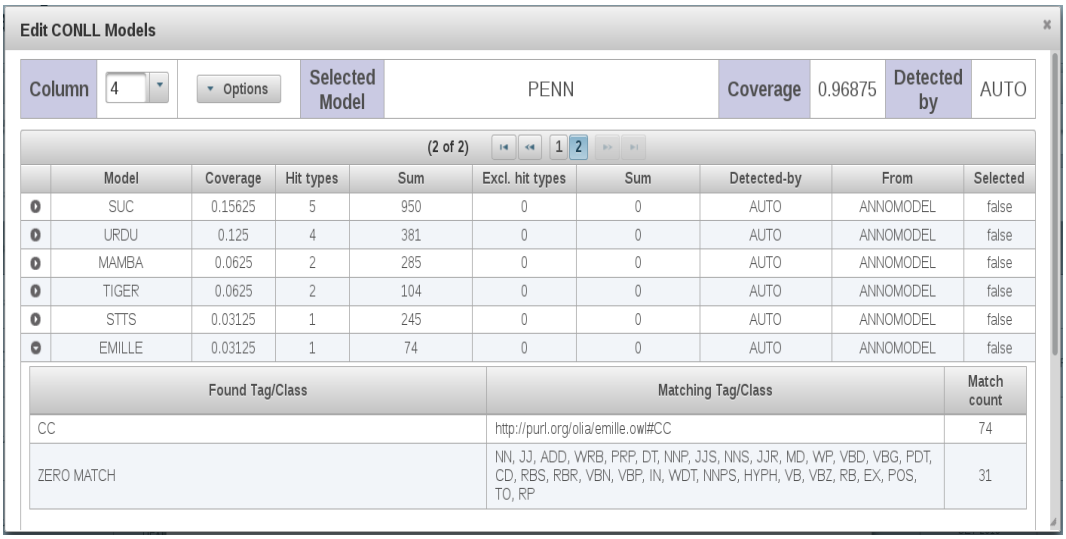


Figure 3 Edit details for the CoNLL file en_ewt-ud-train.conllu.

5 Results

Detailed results are shown in Tab. 3: The first three table columns comprise the results for language detection and the last four columns show the predicted annotation models (were C-4, C-5, C-6 and C-8 refer to the respective columns of a CoNLL file). For table column C-5 a restrictive filtering was applied to the detection results. It only shows those annotation models that could provide a coverage of more than 95%. The UD columns have coverage scores of 100% (C-4, C-6, UPOS and DEPS), resp., 83% – 100% (C-8, FEATS).

As an example consider the file ar_padt. In column 2 (WORD), the language was detected for which the corpus was also marked in the metadata (ara, Macro-Arabic [all varieties]), in column 3 (LEMMA), a different variety was predicted (arb, Standard Arabic), counting here as an error. However, as the first tag was correct, this counts as a partial match.[15] In column 4, the UD part-of-speech, in column 6 the UD features and in column 8 UD dependency labels were detected. Column 5 did not produce an annotation model with coverage greater than 95%. Columns 9 and following were generally excluded.

6 Summary and Outlook

We presented a method to analyze and to curate language resources with respect to their annotation schemes and languages. This functionality is provided as a component to faciliate for metadata indexing and search functionalities in an information system tailored for applications in the language sciences, where it will be applied to provide search beyond bibliographical references to relevant language resources. We specifically described the treatment of TSV formats as frequently used for corpora and provided an evaluation against the Universal Dependencies corpora. We are currently in the process of extending the described methods to CoNLL and TSV formats beyond the Universal Dependencies. In

[15] Of course, this is most likely not an error, as Standard Arabic is a variety of Arabic. But we ground our evaluation in the available metadata.

Table 3 Detailed prediction results for 93 UD corpora.

	Languages			Annotation Models			
corpus	**ISO 639-3**	**predicted**	**comment**	**C-4**	**C-5**	**C-6**	**C-8**
ar_nyuad	—	—	✓(no text)	UD	†	UD	UD
ar_padt	ara	ara,arb	partial match	UD	†	UD	UD
bxr_bdt	bxr	khk	no profile	UD	—	UD	UD
ca_ancora	cat	cat	✓	UD	UD	UD	UD
cop_scriptorium	cop	—	fail	UD	†	UD	UD
cu_proiel	chu	bul	no profile	UD	†	UD	UD
de_gsd	deu	deu	✓	UD	STTS	UD	UD
el_gdt	ell	ell	✓	UD	UD	UD	UD
en_esl	—	—	✓(no text)	UD	PENN	—	UD
en_ewt	eng	eng	✓	UD	PENN	UD	UD
en_gum	eng	eng	✓	UD	PENN	UD	UD
es_ancora	spa	spa	✓	UD	UD	UD	UD
fr_ftb	—	—	✓(no text)	UD	—	UD	UD
fro_srcmf	fro	fra	no profile	UD	†	UD	UD
fr_spoken	fra	fra	✓	UD	—	—	UD
gl_ctg	glg	glg	✓	UD	†	—	UD
grc_perseus	grc	grc,ell	partial match	UD	†	UD	UD
he_htb	heb	heb	✓	UD	UD	UD	UD
hi_hdtb	hin	hin	✓	UD	ANCORRA	UD	UD
hsb_ufal	hsb	pol	no profile	UD	—	UD	UD
ja_bccwj	—	—	✓(no text)	UD	—	—	UD
kk_ktb	kaz	bel	no profile	UD	†	UD	UD
ko_gsd	kor	kor	✓	UD	†	—	UD
ko_kaist	kor	kor	✓	UD	†	—	UD
no_nynorsklia	nor	nno,nob	weak match	UD	†	UD	UD
no_nynorsk	nor	nno	weak match	UD	—	UD	UD
ro_rrt	ron	ron	✓	UD	MULT	UD	UD
ru_gsd	rus	rus,bul	partial match	UD	PENN	UD	UD
sk_snk	slk	slk	✓	UD	MULT	UD	UD
sl_ssj	slv	slv	✓	UD	MULT	UD	UD
sl_sst	slv	slv	✓	UD	MULT	UD	UD
sme_giella	sme	prf	no profile	UD	†	UD	UD
sr_set	srp	hrv	fail	UD	—	UD	UD
swl_sslc	swl	ude	no profile	UD	†	—	UD
te_mtg	tel	tel	✓	UD	UD	UD	UD
ug_udt	uig	pes	no profile	UD	†	UD	UD
uk_iu	ukr	ukr	✓	UD	MULT	UD	UD
zh_gsd	zho	zho	✓	UD	PENN	UD	UD
55 other[1)]	correct		✓	UD		UD	UD

1) 55 corpora for 39 languages, afr, bel, bul, ces, dan, eng, est, eus, fas, fin, fra, gle, glg, got, grc, hrv, hun, hye, ind, ita, jpn, kmr, lat, lit, lav, mar, mlt, nld, nor, pol, por, ron, rus, spa, swe, tam, tur, urd, vie († marks a model coverage < 96%, — marks no language or model info in corpus)

addition, other corpus and dictionary formats will be supported, most noteably various XML formats. A generic XML converter/indexer is currently under development. The code of our detectors and the generated RDF metadata from this resources will be published in early 2020 under an open license[16].

[16] `https://annohub.linguistik.de`

References

1 Christian Chiarcos and Christian Fäth. CoNLL-RDF: Linked corpora done in an NLP-friendly way. In Jorge Gracia, Francis Bond, John P. McCrae, Paul Buitelaar, Christian Chiarcos, and Sebastian Hellmann, editors, *Language, Data, and Knowledge*, pages 74–88, Cham, Switzerland, 2017. Springer.

2 Christian Chiarcos, Christian Fäth, Heike Renner-Westermann, Frank Abromeit, and Vanya Dimitrova. Lin|gu|is|tik: Building the Linguist's Pathway to Bibliographies, Libraries, Language Resources and Linked Open Data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, 2016. European Language Resources Association (ELRA).

3 Christian Chiarcos, Sebastian Nordhoff, and Sebastian Hellmann. *Linked Data in Linguistics*. Springer, 2012.

4 Christian Chiarcos and Maria Sukhareva. OLiA – Ontologies of Linguistic Annotation. *Semantic Web Journal,518:379–386*, 2015.

5 Christian Chiarcos, Maria Sukhareva, Roland Mittmann, Timothy Price, Gaye Detmold, and Jan Chobotsky. New Technologies for Old Germanic. Resources and Research on Parallel Bibles in Older Continental Western Germanic. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 22–31. Association for Computational Linguistics, 2014.

6 Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, 2016. European Language Resources Association (ELRA).

7 Andrea Zielinski and Christian Simon. Morphisto – An Open Source Morphological Analyzer for German. In *Proceedings of the 2009 Conference on Finite-State Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP 2008*, pages 224–231, Amsterdam, The Netherlands, 2009. IOS Press.

The Secret to Popular Chinese Web Novels: A Corpus-Driven Study

Yi-Ju Lin
Graduate Institute of Linguistics, National Taiwan University, Taiwan
jl8394@gmail.com

Shu-Kai Hsieh
Graduate Institute of Linguistics, National Taiwan University, Taiwan
shukai@gmail.com

Abstract

What is the secret to writing popular novels? The issue is an intriguing one among researchers from various fields. The goal of this study is to identify the linguistic features of several popular web novels as well as how the textual features found within and the overall tone interact with the genre and themes of each novel. Apart from writing style, non-textual information may also reveal details behind the success of web novels. Since web fiction has become a major industry with top writers making millions of dollars and their stories adapted into published books, determining essential elements of "publishable" novels is of importance. The present study further examines how non-textual information, namely, the number of hits, shares, favorites, and comments, may contribute to several features of the most popular published and unpublished web novels. Findings reveal that keywords, function words, and lexical diversity of a novel are highly related to its genres and writing style while dialogue proportion shows the narration voice of the story. In addition, relatively shorter sentences are found in these novels. The data also reveal that the number of favorites and comments serve as significant predictors for the number of shares and hits of unpublished web novels, respectively; however, the number of hits and shares of published web novels is more unpredictable.

2012 ACM Subject Classification General and reference → Empirical studies; General and reference

Keywords and phrases Popular Chinese Web Novels, NLP techniques, Sentiment Analysis, Publication of Web novels

Digital Object Identifier 10.4230/OASIcs.LDK.2019.24

Category Short Paper

1 Introduction

Is there a common pattern for popular novels? This is a curious question among publishers, professional book reviewers, and even researchers from various fields. More recently, with an increase in employment of empirical methods in studies of linguistics, exploiting and combining computational tool into research of language and literature has increasingly been the object of study in recent years.

The goal of this research is to identify the textual properties along with the external factors that may contribute to popular web novels in Taiwan. As previous literature indicated [1, 6, 9, 11], stylistic features are essential in differentiating authorial style and text genre. The first part of this study examines the textual content of these popular online novels. In other words, several stylistic features and the overall sentiment tone of the top 3 hit novels were investigated by exploiting Natural Language Processing (NLP) techniques such as keyword extraction and sentiment analysis. The most prominent features that can discriminate different genres and styles the best, for example, high-frequency words, dialogue proportion, average sentence length, and lexical richness, were displayed to show the shared textual elements in popular web novels. Finally, as previous literature [3, 19] noted, examining


© Yi-Ju Lin and Shu-Kai Hsieh;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 24; pp. 24:1–24:8

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

writing style alone does not define a novel's success. It has to do a lot with book promotion and reader's feedback. In this paper, the non-textual information such as the number of hits, shares, favorites, and comments was examined to predict the top hit web novels and published web novels. The study attempts to expand the understanding of the shared elements among popular Chinese web novels in Taiwan.

2 Related Work

The anatomy of successful literary works is an intriguing issue among publishers and aspiring writers alike, and even researchers from various fields. Related works on the stylistic features of literary works are abundant. A number of publications [2, 5, 6, 9] have focused on the stylistic aspects in characterizing different genres and styles of literature. It is showed that using linguistic cues in classifying genres of literary works is effective. These discriminating features include passive use, terms of address (e.g., Mr., Ms.), frequent words, punctuation cues, dialogue proportion. In addition, syntactic features are said to be helpful in distinguishing genres of novels. Juatze (2013) [8] reported that literature contains more complex (e.g., subordinating) sentences than chick literature (humorous novels on the challenges of being a modern-day urban female). However, it is worth noting that none of these previous works were done on language other than English. Therefore, whether these prominent stylistic markers shown in previous literature are also effective in differentiating genres and styles of Chinese novels is worth discussing. The work of Yu (2012) [18] pointed out the effectiveness of using function words for Chinese authorship attribution in different genres. It is also noted by Wu (2017) [17] that stylistic features such as average sentence length and vocabulary pattern are able to differentiate authorial styles.

A few studies were carried out on the quantitative connection between writing style and successful literature. Ganjigunte Ashok et al (2013) [3] revealed that there exist distinct linguistic patterns shared among successful literature. It indicated that popular novels use lots of conjunctions, while less popular books use more verbs, adverbs and foreign words. The groundbreaking study then built a model with surprisingly high accuracy (up to 84%) in predicting the success of a novel by using statistical methods.

From the growing body of literature on applying sentiment analysis techniques to the text of fiction, it seems clear that using sentiment analysis for understanding fiction emotion provides another way of analyzing the genres and writing style of fiction. Sentiment Analysis is, at present, widely applied in the areas of product and movie reviews (Hu et al 2004 [7]; Sreejith et al 2017 [14]), whereas for this paper, we have tried to use it in longer texts like novels. Landt (2010) [10] examined English fan-translated version of the demo of the Japanese visual novel to discover the overall tone of the text, including how the tone changed as the story progressed as well as how sentiment was used to portray the various characters. Landt's (2010) study is useful since the overall tone of the top 3 hit novel was examined in the paper to analyze the relationship between the overall tone of fiction and its theme.

According to [16, 19], examining textual content alone is not enough in determining the popularity of a book. It is showed that non-textual information such as sales number, online reviews, and readers' interactive feedback on media platform also play a role in book success. However, previous studies focused mainly on printed books. Little effort has been devoted to combining online rating into determining the success of web novels.

3 Methodology

3.1 Data and Sample Selection

There is a growing base of web fiction offered for free. The novels analyzed in this study were all extracted from a free online novel website, Mirror fiction. The site is one of the best platforms in Taiwan currently offering original stories online. Some of the website top stories have received several millions of views. Established in 2017, Mirror fiction aims to create a platform which enables more creators' works to be officially published, authorized and adapted into published books, films and television works. Over hundreds of categories of fiction are listed on the site, including fan fiction, mystery, romance and thriller. . . etc. Readers are able to keep, share, and comment on the works.

Our corpus consists of 9 top hit novels of different genres. The rankings used here was based on the information released by Mirror fiction website. Using text mining techniques such as keyword extraction and sentiment analysis, the textual content of 9 web novels were analyzed. Furthermore, non-textual information (number of hits, share, and comment) of 59 web novels was investigated to predict top hit and top shared web novels. In addition, the discriminating features between printed web novels and web novels were also explored.

3.2 Text Analysis Tool and NLP techniques

A web-based text analysis tool, HTML5 Text Analyzer, was utilized in the study to identify stylistic features of web novels. The site provides detailed statistics of your text such as lexical richness, sentence length, function word proportion, dialogue proportion, and punctuation proportion. The prominent features that can most depict the genres, plot and themes of the web novels were chosen and presented in the paper.

NLP techniques are utilized to characterize the writing style of top hit novels as well as its interaction with different genres. Term frequency counts, as a way of keyword extraction, were computed for the top 3 hits and 3 different genres of novels(top 2 hits are extracted in each genre). The term frequency list displays the term frequency counts after removing stop words and unrelated terms (e.g. character names). Another way of extracting keywords is by using TF-IDF. It assumes words with high word frequency in the given documents and low document frequency in the whole collection of documents are of high importance. This way of extracting keywords enables higher discrimination power between documents. For example, common words like "the", "of" and "a", which appears in many novels, will be scaled down. Words that frequently appear in few novels, like "firm", "painted scroll", "dating" for indicating the plot of a particular novel, will be scaled up. Furthermore, the current study attempts to find the tone of these top hit novels by applying sentiment analysis. Python package such as Jieba and SnowNLP were used for processing Chinese text. Jieba help segmentation of the novels' text while SnowNLP analyzes the sentences of a novel and outputs sentiment score that indicates the probability of showing positive emotion.

In [19], the author highlighted the importance of combining readers' online ratings and reviews into determining popular fictions. The numerical statistics on Mirror fiction may reveal something about the success of novels. Therefore, 7 unpublished web novels and 11 published web novels were selected from the website. The number of hits, favorites, shares, and comments of these novels were selected as predictors to predict the factors that make web novels "publishable" by using regression model under statistical software R. Notice that these numerical variables (e.g., number of hits, shares, comments, and favorites) indicated the popularity before the novels have been published into hardcover.

4 Results and Discussions

4.1 Textual properties of top hit novels

What are the lexicon that is most frequently used in top hit novels? The most frequently used words of the top 3 hits were extracted and analyzed. In general, the frequency list is able to depict the genre and setting of the story by only looking at the top 10 frequent terms. For example, *Ghost Mansion*, one of the top hit novels extracted from Mirror fiction website, often uses terms like gong si 'firm', gong zuo 'work' , nu ren 'women' in the story. These frequent terms picture the setting of the novel, which focuses on the social lives and relationships of young professional women. Such kind of settings is often being categorized as "chick literature" which tells the story of the personal growth of a woman or deals with modern issues in women's lives.

Additionally, tf-idf algorithm was used in this study. In is found that top 10 terms of top 3 hits ranked by tf-idf give a more accurate depiction on the story's genre and setting. However, it must be noted that since the algorithm gives higher weights to terms that are common in one document but unique among all others documents, character's name is given higher weights in the tf-idf list. The issue, however, has been resolved by removing character's names. The algorithm showed a clearer picture of the setting of the novel. In chick literature, terms like xin wen 'news', zhu bo 'anchor', qi hua 'marcom', gong zuo 'work' that related to the modern issues in women's lives appear on the list. On the other hand, expressions in classical Chinese like shi fu 'master', gong zhu 'princess', ming yue 'bright moon' picture the setting of historical novel.

Stylistic features provide another way of characterizing writing style of top hit novels. It is also believed that styles of the novel can be distinguished along certain textual features (Argamon 2006[2]; de Haan 1997[4]; Juatze 2013[8]). Table 1 shows the features that can most depict the themes and topic of top hit novels. Table 2 demonstrates the most discriminating features in different genres. *The Mantra* shows the lowest Simpson's Index, which means it has the highest lexical richness. Simpson's D [1](Simpson 1949, as presented in Tweedie & Baayen, 1998[15]) is calculated by:

$$D = \sum_{i=1}^{V} f_v(i, N) \frac{i}{N} \frac{i-1}{N-1}$$

where N refers to the total number of tokens, V to the number of types, and fv(i,N) to the numbers of types occurring i times in a sample of length N. I interpret this to mean that diverse vocabularies are used in *The Mantra* (historical fiction) to help the reader get immersed in the historical events and settings which are farther away from normal people's real life.

Jodie Archer and Matthew L. Jockers' *The Bestseller Code: Anatomy of the Blockbuster Novel*(2016)[1] showed that readers of bestsellers liked shorter sentences. It is shown in our data that popular web novels have shorter sentences than others, since the average length of Chinese novel is around 23 words per sentence [17]. Furthermore, interesting findings were revealed, showing that historical novels have the shortest sentence compared to the other two types of novel. This is reasonable since most of the historical novels are written in a mixture of modern vernacular Chinese and written classical Chinese, a traditional style of written Chinese that appears extremely concise and compact compared to modern spoken form of Chinese.

[1] A measure of lexical richness, calculate the frequency of different words in the writings. The smaller the value, the higher the lexical richness

As de Haan (1997)[4] noted, dialogue plays a part in differentiating genres of fiction texts. The proportion of dialogue and narrative will vary depending on the story's setting and genre. Our findings echoed with de Haan's (1997) study in some way. Furthermore, it is also found that dialogue is related to narrative voice(the format through which a story is communicated) of the story. As observed from Table 1, a relatively high proportion of dialogue is used in *My Heart Belongs to You*(Romance), while *Ghost Mansion*(chick literature) showed a low percentage of dialogue. The result could be interpreted in two ways. First, higher proportion of dialogue is used as a strategy in third person narration novel like *My Heart Belongs to You* to clarify the complicate relationship between characters while a first-person viewpoint story such as *Ghost Mansion* requires much less. Second, romance like *My Heart Belongs to You* requires a lot of dialogue because the relationship among the male and female characters is complex while a chick literature such as *Ghost Mansion* involves only characters in workplace.

Function words are said to be effective in distinguishing different writing styles and genres of novels [18]. However, as shown in Table 1, function words are not the distinguishing features in discriminating different writing style. Similarly in Table 2, function words are not able to discriminate historical novels and Romance.

The lexical choice and stylistic features, however, cannot depict the tone and emotion embedded in the story. As some researchers (Landt 2010 [10]; Sreejith et al 2017 [14]) argued, sentiment analysis of literary works is a useful tool in analyzing fiction. It is further highlighted in Landt's (2010) study that more research should be done on the interaction between the genre of text and its overall tone. It is revealed in this study that the overall tone of the text is highly related to its genres and themes. For example, *Ghost Mansion* has a lower overall sentiment score. This is due to the fact that the theme of the story is about the collusion between politicians and real estate tycoons. Although the story is categorized as a chick literature, the major part of the story is exposing the sordid underbelly of modern urban society. *My Heart Belongs to You*(romance) has the highest overall sentiment score among the three books. As a typical romance novel, there might be conflict that hinders the couple's relationship, but romance is still the overriding element in this kind of story. This explains the fact that the novel has the most positive overall sentiment.

4.2 What can numbers reveal about the success of web novels

Apart from the writing style, there are multiple factors that can determine the success of web novels. First, identifying popular tags in different genres of novel reveals readers' preference on the "topic" of the story. Our findings indicated that readers prefer topics on "urban", "workplace", and "modern". As shown in our analysis, the tags "modern", "urban", "workplace" are the most popular tags in chick literature. This result is highly in line with the main idea of chick literature, which often addresses issues of modern womanhood – from romantic relationships to female friendships to matters in the workplace. It is also showed that "time travel" is the most popular tags in historical novels. This is clearly related to the emergence of a novel genre called "alternative history" [13], which the protagonist (mostly women) travels from modern China to ancient China. "Urban" is showed to be the most popular tag in romance fiction writing. Adding "urban" flavor into romance novels make the story close to readers' real world since romance is mainly marketed to middle-class women who live in the urban area (Radway1991 [12]).

Web fiction has becoming a major industry with top writers making millions of dollars and their stories adapted into published books, TV, and movies. Therefore, determining essential elements of "publishable" novels is of importance. First of all, t-test analysis was conducted to evaluate the hypothesis that there is a significant difference in the number of

hits, shares, comments, and favorites of unpublished and published web novels. The test was significant only on the number of favorites, t (16) = -2.7079, $p<0.05$. In other words, the number of favorites of published web novel is significantly greater than unpublished web novels. An explanation for this is that readers prefer to add the novel to "favorites" and download it before it is being published into paper book. This is reasonable since once the novel has been published, the reader is unable to view the book for free anymore.

Next, multiple regression analysis was then conducted to discover what factors result in a higher number of shares based on the number of favorites, hits, and comments (Table 3). In the group of unpublished web novels, our model shows that the number of favorites is the only significant predictor to explain the increase in the number of share. As for published web novel, the number of comments is the only predictor among the three variables that is able to explain the increase in the number of shares. We can infer from the result that for unpublished novels, whether it can be shared on social media (Facebook, Instagram, Twitter. . . etc.) depends heavily on the number of favorites from the website. On the other hand, for published web novels, the result is more apparent which shows that more comments lead to more shares on social media.

Finally, another multiple linear regression model was built to determine which variables contribute to the top hit of novels (Table 4). For unpublished web novels, the number of comments is prominent in predicting the number of hits. However, the situation becomes more complicated when it comes to predicting the number of hits in published web novel. There are no variables that can explain the number of hits in published web novels. One possible explanation for this is that once web novel has been published into paper book, multiple factors play in the role of the increase in the number of hits on the website. The factor that can mostly explain the number of hits is beyond the variables investigated in this study.

5 Conclusion

In this paper, the textual and non-textual features of popular web novels written with traditional Chinese have been analyzed. First, from examining the textual content of the most popular novels, it is evident that certain features such as keywords, function words and lexical diversity of the novel are highly related to the genres and writing style of the novel while dialogue proportion reveals something about the narrative mode of the story. Additionally, it is found that shorter sentences are favored by readers on Mirror fiction. The general sentiment in the novel is closely linked to the genre and themes of the story. This result is in line with Landt's (2010) [10] study, although no previous study had dealt with the issue in detail. Finally, the data reveal that the number of favorites and comments serve as significant predictors for the number of shares and hits of unpublished web novels, respectively. However, the number of hits and shares of published web novels is more unpredictable.

The current study makes an attempt to discover how the NLP techniques can help to explain popular web novels in Taiwan . However, since the study involved only "popular novels" but no "less popular novels", the discriminating features between highly popular ones from less popular ones cannot be determined. Another limitation concerns the sample size. Data collected in our research is too small to make a more accurate generalization on the writing styles among different genres. Given the exploratory nature of this study, it is hoped that it can serve as a basis for further study in exploring the secret to popular web novels in Taiwan.

References

1 Jodie Archer and Matthew L. Jockers. *The Bestseller Code: Anatomy of the Blockbuster Novel*. St. Martin's Press, Inc., New York, NY, USA, 2016.

2 Shlomo Argamon, Moshe Koppel, Jonathan Fine, and Anat Rachel Shimoni. Gender, genre, and writing style in formal written texts. *Text - Interdisciplinary Journal for the Study of Discourse*, 23:321–346, 2006.

3 V.G. Ashok, S Feng, and Y Choi. Success with style: Using writing style to predict the success of novels. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1753–1764, 2013.

4 Pieter de Haan. More on the language of dialogue in fiction. *ICAME Journal*, 20, 1997.

5 Helena Montserrat Gomez Adorno, Germán Rios, Juan Pablo Posadas Durán, Grigori Sidorov, and Gerardo Sierra. Stylometry-based Approach for Detecting Writing Style Changes in Literary Texts. *Computación y Sistemas*, 22(1), 2018.

6 David L. Hoover. Frequent Collocations and Authorial Style. *Literary and Linguistic Computing*, 18(3):261–286, 2003. doi:10.1093/llc/18.3.261.

7 Minqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, New York, NY, USA, 2004. ACM. doi:10.1145/1014052.1014073.

8 Kim Jautze, Corina Koolen, Andreas van Cranenburgh, and Hayco de Jong. From high heels to weed attics: a syntactic investigation of chick lit and literature. In *Proceedings of the Workshop on Computational Linguistics for Literature*, pages 72–81. Association for Computational Linguistics, 2013. URL: http://aclweb.org/anthology/W13-1410.

9 Brett Kessler, Geoffrey Nunberg, and Hinrich Schütze. Automatic Detection of Text Genre. In *Proceedings of ACL-97, 35th Annual Meeting of the Association for Computational Linguistics*, pages 32–38, 1997.

10 Matthias Landt. Sentiment Analysis as a Tool for Understanding Fiction. In *ACM 2010 Annual Meeting*, 2010.

11 Ying Liu and TianJiu Xiao. A Stylistic Analysis for Gu Long's Kung Fu Novels. *Journal of Quantitative Linguistics*, pages 1–30, 2018. doi:10.1080/09296174.2018.1504411.

12 Jeanice A. Radway. *Reading the Romance: Women, Patriarchy, and Popular Literature*. University of North Carolina Press, 1991. URL: http://www.jstor.org/stable/10.5149/9780807898857_radway.

13 Biwu Shang. Unnatural narratives in contemporary Chinese time travel fiction: patterns, values, and interpretive options. *Neohelicon*, 43:7–25, July 2016. doi:10.1007/s11059-016-0327-z.

14 D. Sreejith, M. P. Devika, Naga Santosh Tadikamalla, and Sanju Varghese Mathew. Sentiment Analysis of English Literature using Rasa-Oriented Semantic Ontology. *Indian Journal of Science and Technology*, 10(24), 2017. URL: http://www.indjst.org/index.php/indjst/article/view/96498.

15 Fiona J. Tweedie and R. Harald Baayen. How Variable May a Constant be? Measures of Lexical Richness in Perspective. *Computers and the Humanities*, 32(5):323–352, September 1998. doi:10.1023/A:1001749303137.

16 Marc Verboord. Cultural products go online: Comparing the internet and print media on distributions of gender, genre and commercial success. *Communications*, 36(4):441–462, 2011.

17 Chin-Wei Wu. A Linguistic Stylistic Analysis of the Sentences in Wang Wen-hsing's Novel–Backed Against the Sea. *Journal of Chinese Literature of National Cheng Kung University*, 59:181–215, 2017. doi:10.1016/j.dcm.2018.03.003.

18 Bei Yu. Function Words for Chinese Authorship Attribution. In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 45–53, Montréal, Canada, June 2012. Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/W12-2506.

19 Burcu Yucesoy, Xindi Wang, Junming Huang, and Albert-László Barabási. Success in books: a big data approach to bestsellers. *EPJ Data Science*, 7(1), April 2018. doi:10.1140/epjds/s13688-018-0135-y.

A Tables

Table 1 Statistics on the top 3 most popular novels.

	Text Length	Simpson's Index	Average Sentence Length	Dialogue Proportion	Function Word Proportion
Ghost Mansion	91084	0.0003	10.9383	0.05841	0.4836
The Mantra	94446	0.00022	7.31742	0.11776	0.4268
My Heart Belongs to You	45586	0.00078	10.4101	0.18222	0.4775

Table 2 Statistics on novels of different genres.

	Text Length	Simpson's Index	Average Sentence Length	Dialogue Proportion	Function Word Proportion
Chick Lit	167875	0.004882	10.96362	0.0700819	0.1007
Historical	160155	0.000044	7.32204	0.1407698	0.4303
Romance	58903	0.000245	9.916329	0.1488718	0.4904

Table 3 Regression results for predicting the number of shares with number the of favorites, hits, and comments.

Coefficients				
	Unpublished Web novel		Published web novel	
Independent variable	t value	p	t value	p
favorites	6.996	**.00599****	0.848	0.199
hits	-1.162	0.32911	1.197	0.27
comments	0.851	0.45742	-1.309	**.026***
	Adjusted R^2 : 0.9257, $p < 0.05*$		Adjusted R^2 : 0.7313, $p < 0.01**$	
significant values: *p<0.05, **p < 0.01, ***p< 0.001				

Table 4 Regression results for predicting the number of hits with the number of favorites, shares, and comments.

Coefficients				
	Unpublished Web novel		Published web novel	
Independent variable	t value	p	t value	p
favorites	1.374	0.26315	0.848	0.424
shares	-1.162	0.32911	1.197	0.27
comments	6.205	**.00844****	-1.309	0.232
	Adjusted R^2 : 0.9506, p<0.01**		Adjusted R^2: 0.289, p=.1578	
significant values: *p<0.05, **p <0.01, ***p<0.001				

Predicting Math Success in an Online Tutoring System Using Language Data and Click-Stream Variables: A Longitudinal Analysis

Scott Crossley
Georgia State University, Applied Linguistics/ESL, Atlanta, GA, USA
scrossley@gsu.edu

Shamya Karumbaiah
The University of Pennsylvania, Philadelphia, PA, USA
shamya@upenn.edu

Jaclyn Ocumpaugh
The University of Pennsylvania, Philadelphia, PA, USA
ojaclyn@upenn.edu

Matthew J. Labrum
Imagine Learning, Provo, UT, USA
matthew.labrum@imaginelearning.com

Ryan S. Baker
The University of Pennsylvania, Philadelphia, PA, USA
rybaker@upenn.edu

Abstract

Previous studies have demonstrated strong links between students' linguistic knowledge, their affective language patterns and their success in math. Other studies have shown that demographic and click-stream variables in online learning environments are important predictors of math success. This study builds on this research in two ways. First, it combines linguistics and click-stream variables along with demographic information to increase prediction rates for math success. Second, it examines how random variance, as found in repeated participant data, can explain math success beyond linguistic, demographic, and click-stream variables. The findings indicate that linguistic, demographic, and click-stream factors explained about 14% of the variance in math scores. These variables mixed with random factors explained about 44% of the variance.

2012 ACM Subject Classification Applied computing → Computer-assisted instruction; Applied computing → Mathematics and statistics; Computing methodologies → Natural language processing

Keywords and phrases Natural language processing, math education, online tutoring systems, text analytics, click-stream variables

Digital Object Identifier 10.4230/OASIcs.LDK.2019.25

Funding This research was supported in part by NSF 1623730. Opinions, conclusions, or recommendations do not necessarily reflect the views of the NSF.

1 Introduction

Students need a number of cognitive skills including spatial attention and quantitative ability to be successful within a math classroom [28]. In addition, recent research has shown strong links between students' language production and math success. This research demonstrates that students that are more proficient in math are generally also more proficient language users. There are several potential reasons for links between math and language domains, both of which rely on the ability to interpret and manipulate abstract symbolic systems [40]. One


© Scott Crossley, Shamya Karumbaiah, Jaclyn Ocumpaugh, Matthew J. Labrum, and Ryan S. Baker; licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 25; pp. 25:1–25:13

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

key reason is that language skills help students learn knowledge and math operations from text books and tutoring systems, as well as from other people. More generally, students with greater language proficiency are better able to engage individually and collaboratively with math concepts and solve math problems because math is not purely based on numbers and abstract symbols but also on real world problems that involve the words surrounding math numbers and symbols [2]. Thus, language skills help students to participate constructively and collaboratively in math discourse and engage with and solve math problems both inside and outside of the classroom [30, 41].

Some previous studies that have examined links between math success and language have relied on correlational analyses between standardized tests of language and math. As an example, previous studies have analyzed association between language proficiency tests that assess syntax, lexical, and phonological skills and math scores on standardized tests that assess arithmetic and algebra and found strong links [29, 41]. Another area of inquiry between language proficiency and math skills has been to compare success rates on standardized math tests between native and non-native speakers of English. These studies often find that non-native speakers of English perform lower on math assessments [3, 20, 31] although see [1] for counter argument. A final approach to examining math and language links is to assess links between the complexity of language produced by students and their success on math assessments. Such studies generally find that students that produce more complex language features score higher in math, possibly because students' ability to switch from conversational language to the conventions required in mathematics requires high level metalinguistic skills [19].

The current study builds on previous studies that have focused on links between the language produced by students and their math success, by combining fine-grained click-stream variables and simple demographic data (i.e., grade and gender) with language features in student production to predict math success. We also assess math performance over time to better control for variance associated with participants. To do so, we use natural language processing (NLP) tools to assess language production in e-mail messages sent by elementary students within an online tutoring system over the course of a year. We then examine the students' behaviors within the tutoring system in terms of actions completed, entries into various elements of the system, and time spent within these elements. The goals of the study are to combine these data to increase prediction rates within the system. Additionally, we examine performance over time to assess the degree to which random variance found in repeated participant data can explain math success beyond linguistics, demographic, and click-stream data.

1.1 Relationships between Language and Math Skills

The body of research demonstrating connections between proficiency in language and math skills continues to grow, becoming more robust as researchers explore the potential underlying causes. Early studies focused on links between scores on math and language tests. For instance, MacGregor and Price [29] found that students who scored high on an algebra test also scored well on language tests. Using a more difficult algebra test produced a stronger relationship between algebraic notation and language ability. Similarly, Hernandez [22] found significant positive correlations between reading and math scores in standardized tests. Vukovic and Lesaux [41] also reported links between language and math skills, but additionally found that language skills differed in their degree of relation with math knowledge. For example, general verbal ability was indirectly related with symbolic number skills while phonological skills were directly related to arithmetic knowledge. Lastly, LeFevre et al. [28] reported that language ability was positively related to number naming.

More recent studies have begun to examine links between the language features found in students' language production and their success in math learning using NLP tools. These studies have focused on elementary and college level students. In an early study of elementary students, Crossley et al. [12] examined language features found in transcribed student speech during collaborative math projects and found that language features related to cohesion, affect, and lexical proficiency explained a significant amount of variance in students' math scores. More mathematically proficient students produced more cohesive language that was comprised of more lexically sophisticated words. In another study, Crossley and Kostyuk [11] examined links between the language features of elementary students' language production while e-mailing a virtual pedagogical agent in an online math tutoring system and math success within the system. They found that students who expressed more certainty in their writing and followed standardized language patterns scored higher in math assessments. In a more recent study, Crossley et al. [13] used linguistic features found in student e-mails within an online math tutoring system to predict math success. They found that lexical features and syntactic complexity indices were significant predictors of math success such that more successful students used words that were found across a variety of registers and used more sophisticated words. In addition, higher scoring math students produced fewer complex sentences.

Studies assessing links between language features and math success for college level students have reported similar findings. For instance, Crossley et al. [10] examined college students' forum posts in an online tutoring system that was part of a blended math class (i.e., a class with both online and traditional face-to-face instruction). They investigated relationships between language features in these posts and final scores in the class, finding that success in the class was predicted by language features related to affect, syntactic complexity, and text cohesion. Specifically, more complex syntactic structures and fewer explicit cohesion devices were associated with higher course performance. The linguistic model also indicated that less self-centered students and students using words related to tool use were more successful. In a similar study using the same data set, Crossley et al. [16] examined how linguistic features derived from cohesion network analyses could predict math success. The models from this study indicated that students who encouraged greater language collaboration within forum posts (i.e., those students that precipitated discussion among other students) received higher final scores in the class.

In general, these studies demonstrate that linguistic features from students' language can predict math performance across grade levels (from elementary to college level students) in different types of learning environments (collaborative online tutors, traditional online tutoring systems, and blended math courses). Overall, younger students that are more proficient at math produce more cohesive language that includes more sophisticated vocabulary. In addition, younger students that are more proficient at math are better at following expected language patterns and produce less complex syntactic structures. In contrast, older students who receive higher grades in math class produce more syntactically complex structures that are less cohesive. These students also encourage greater collaboration through their language use. The differences between older and younger students' language production is likely related to different stages of language acquisition.

1.2 Click-stream Data and Student Success in Math

There is growing research that demonstrates the strength of using student interaction data [34] in online learning environments (i.e., click-stream data) to predict short- to long-term learning, engagement and interest in mathematics. Data on fine-grained aspects of student

behavior provides opportunities to explore how patterns of interaction relate to outcomes. For instance, Beal et al. [8] reported that students' use of interactive multimedia hints in an online tutor for SAT-Math problems were predictive of learning gains in the system. To extract richer information from the raw student log data, researchers have also extensively used student interaction data for a discovery with models approach [23]. For example, student interactions in a math tutor were used to build a predictive model of students' careless errors [38], and those models were connected with predictive models of affective states to study the relationship between affective states and carelessness [39].

Log data from math tutors have also been used to predict student scores on end-of-year state accountability exams, resulting in better prediction than paper-pencil benchmark tests and standardized tests [4, 18, 21]. These models become better still when supplemented with data on student strategies [36]. Xie et al. [42] showed how learning strategies defined by interaction data (e.g., learning from errors, switching to a new topic, and reviewing previously mastered topics) predicted end of semester assessments.

Other research has found that student behavior in math tutors in middle school year are predictive of long term success. For example, San Pedro et al. [37] found that student carelessness, and intentional misuse to complete problems without learning in a middle school math tutor are associated with lower probability of college attendance and STEM major. Similarly, Ocumpaugh et al. [34] conducted a longitudinal study of the relationship between middle school math performance and interaction-based affect detectors with student's vocational self-efficacy and interest. They found that both self-efficacy and interest in high school were negatively correlated with confusion during middle school, but that both were positively correlated with carelessness.

Recent studies have also examined click-stream data and math success, usually in conjunction with NLP tools. For instance, Crossley and Kostyuk [11] reported that elementary students who met more objectives within an online math tutoring system and those that sent fewer messages to a pedagogical agent, performed better on math problems. Crossley et al. [10, 16] reported that college level students that received higher final scores in a blended math class spent more time in a forum that allowed postings between students and teachers and visited the online learning platform more often.

1.3 Current Study

As discussed above, a number of studies have demonstrated strong links between students' linguistic knowledge, their affective language patterns and their success in math. In addition, studies have shown that click-stream variables are important predictors of success in online learning systems. This study builds on this previous research in two ways. First, it combines linguistics and click-stream variables along with demographic information to increase prediction rates for math success. Second, it examines how random variance, as found in repeated participant data, can explain math success beyond linguistic, demographic, and click-stream variables.

To derive our language features of interest, we analyzed the language produced by students sending email messages to a virtual pedagogical agent within an online math tutoring system. We analyzed the language using several Natural Language Processing (NLP) tools in order to extract language information related to text cohesion, lexical sophistication, and sentiment. Our click-stream data was extracted from the online tutoring data and focused on actions within the system, entries into various modes of the system, and temporal data related to time spent in those modes. Demographic data included grade and gender. We collected data from students in two consecutive semesters (fall and spring) allowing us to track performance over time. Thus, in this study, we address the following research question:

Are linguistic and click-stream factors along with participant variance over time significant predictors of math performance in an online tutoring environment over two semesters of study?

2 Method

2.1 Reasoning Mind

We collected data from Reasoning Mind's *Foundations* product, which is a blended learning mathematics program used in grades 2–5. *Foundations* students learn math in an engaging, animated world at their own pace, while teachers use the system's real-time data to provide one-on-one and small-group interventions [32]. The algorithms and pedagogical logic underlying Foundations (previously called Genie 2) are described in detail by Khatchatryan et al. [24].

The main study mode in *Foundations*, called *Guided Study*, consists of a sequenced curriculum divided into objectives, each of which introduces a new topic (e.g., the distributive property) using interactive explanations, presents problems of increasing difficulty on the topic, and reviews previously studied topics. Within *Guided Study*, every student completes problems addressing the basic knowledge and skills required in the objective. These basic problems (known as A-level problems) typically require only a single step to solve and are the lowest of three possible difficulty levels. Students who do well on A-level problems may also proceed to problems of higher difficulty that require two or three steps to solve (e.g., B-level and C-level problems) within the objective. They may also access the higher-level problems in an independent study mode called *Wall of Mastery*. Other modes in *Foundations* allow students to play math games against classmates, tackle challenging problems and puzzles, and use points earned by solving math problems to buy virtual prizes.

Foundations uses animated characters to provide a backstory to the mathematics being learned and to deliver emotional support. The main character is the Genie, a pedagogical agent who encourages students throughout their work in the system. Students are also able to send emails to the Genie. These messages are answered in character by part-time Reasoning Mind employees who reference an extensive biography of the Genie and project a consistent, warm, and encouraging persona, model a positive attitude toward learning, and emphasize the importance of practice and challenging work for success. The Genie email system is a popular component of the system, having received 129,879 messages from 38,940 different students in the 2016–17 academic year.

2.2 Participants

The students sampled in this study were selected from the 34,602 students who used *Foundations* in the 2016–17 academic year. The students were from 462 different schools located in 99 different districts, most of which were located in Texas. We included those students that had attempted A-level problems in both the fall and spring semester. As an additional requirement, these students needed to have written at least 50 words within the Genie email system (the minimum number of words needed to develop a linguistic profile for the students). From the available student data, 1,036 students met these criteria.

2.3 Genie Email Corpus

Our language sample for this analysis consisted of messages sent from the students to the Genie. Because many messages contained only a few words, we aggregated all emails sent by each student to create a representation of an individual student's linguistic activity.

We then implemented data cleaning procedures to reduce the amount of noise in the data. First, all the data was cleaned of non-ASCII characters that could interfere with the NLP tools. Second, all texts were automatically spell-checked and corrected using an open-source Python spelling correction library, in addition to several Python text-cleaning scripts that we developed. Furthermore, several measures were taken to clean the texts, including removing random, non-math symbols such as "#", "@", and "&", as well as omitting repeating words, excessively long words, words with repeating characters, such as "wooorrrddd", and mixed-type words, such as "$word$" (with the exceptions of currencies, percentages, timestamps, and ordinals). Next, all non-dictionary, invalid words were removed from the data. This was accomplished by first checking each word against synonym sets (synsets) in WordNet, and if a match could not be found, then checking if it consisted of all consonants (always invalid), or if any pair of characters (digraph) in the word were invalid in the English language. Words that met either of these two conditions were removed. Lastly, all texts were cleaned of repeating, non-overlapping groups of words, such as "this word this word this word." Only word groups of lengths two, three, and four were removed by this approach.

2.4 Natural Language Processing Tools

We used several NLP tools to assess the linguistic features in the aggregated posts of sufficient length. These included the Tool for the Automatic Analysis of Lexical Sophistication (TAALES) [26], the Tool for the Automatic Analysis of Cohesion (TAACO) [14], the Tool for the Automatic Analysis of Syntactic Sophistication and Complexity (TAASSC) [27], and the SEntiment ANalysis and Cognition Engine (SEANCE) [15]. In addition, we developed specific indices related to topics commonly discussed with the Genie email system using Latent Dirichlet Allocation (LDA). Thus, the selected NLP features consisted of language variables related to lexical sophistication, text cohesion, syntactic complexity sentiment analysis, and topic similarity respectively. The features are discussed in greater detail below.

TAALES. TAALES [26] is a computational tool that is freely available and easy to use, works on most operating systems, affords batch processing of text files, and incorporates over 100s of classic and newly developed indices of lexical sophistication. These indices measure word frequency, lexical range, n-gram frequency and proportion, academic words and phrases, word information, lexical and phrasal sophistication, and age of exposure. For many indices, TAALES calculates scores for all words (AW), content words (CW), and function words (FW). For instance, for word frequency, TAALES reports frequency counts retrieved the SUBTLexus databases [9].

TAALES also reports on a number of word information and psycholinguistic scores derived from the University of South Florida (USF) norms [33], and the English Lexicon Project (ELP) [5] among others. The USF norms are used to calculate the number of associations per word while the ELP is used to calculate many lexical features including the number of orthographic neighbors a word has (i.e., how many words are spelled similarly). Lastly, TAALES reports on type token ratios (TTR) that reference how many unique words are found within a sample.

TAACO. TAACO [14] incorporates a number of classic and recently developed indices related to text cohesion. TAACO has features for content and function words and provides linguistic counts for both sentence and paragraph markers of cohesion. The tool incorporates WordNet synonym sets, latent semantic analysis, and word2vec features.

Specifically, TAACO calculates sentence and paragraph overlap indices and a variety of connective indices.

TAASSC. TAASSC [27] measures large and fined grained clausal and phrasal indices of syntactic complexity and usage-based frequency/contingency indices of syntactic sophistication. TAASSC includes a number of pre-developed fine-grained indices or clausal complexity and phrasal complexity. In addition, TAASCC reports on features related to verb argument constructions (VACs) including the frequency of VACs and the attested constructions in reference corpora taken from the Corpus of Contemporary American English (COCA) [17] to include sub-corpora such as academic writing, magazines, and fiction.

SEANCE. SEANCE [15] is a sentiment analysis tools that relies on a number of pre-existing sentiment, social positioning, and cognition dictionaries. SEANCE contains a number of pre-developed word vectors to measure sentiment, cognition, and social order. These vectors are taken from freely available source databases. For many of these vectors, SEANCE also provides a negation feature (i.e., a contextual valence shifter) that ignores positive terms that are negated (e.g., not happy). SEANCE also includes a part of speech (POS) tagger.

2.5 Click-Stream Variables

Reasoning Mind extensively logs the interaction of the students in the system at the action-level. Actions include logging in, entering a mode, seeing a problem, submitting an answer, and reviewing theory. The type of action a student can take depends on the mode they are in. For instance, in *City Landscape*, students can switch modes such as *Guided Study* and *Wall of Mastery*, where students practice math problems. The *Game Room* also provides an opportunity for students to learn math through games. In contrast, no math learning happens in the *City Landscape* mode, which is simply the landing page in Reasoning Mind from where the student navigates to other modes. Students can also spend time in the *Shopping Mall* purchasing items to decorate their *My Place*, which might have an impact on their overall engagement with the system.

To explore the student interaction patterns in the tutor, we engineered features that captured the distribution of a student's effort in the various phases of the tutor. Along with quantifying the kind of actions a student performs, these features also measure their persistence in an activity. For instance, a higher number of visits to *City Landscape* mode would denote that the student is less persistent in focusing on a single activity.

For each student, we extracted features based on the actions in the 27 modes (e.g., *City Landscape*), the actions within the 11 module types of *Guided Study* mode (e.g., *Introduction*, *Theory*, *Problems*, *Homework*) and 6 content types (e.g., *Problem A*, *Problem B*). For each of these, we mined the log data to extract three kinds of features – 1) number of entries to the mode (e.g., number of entries to *City Landscape* mode); 2) number of actions performed in the mode/module/content type (e.g., number of actions performed in *Problems* module type); 3) total time-spent in the mode/content type (e.g., total time spent solving *Problem A* content type). Thus, the feature *Time in City Landscape* refers to the total sum of time spent by a student across all logins in the *City Landscape* mode. Similarly, the feature *Number of entries to Guided Study* is calculated by counting the number of times a student enters the *Guided Study* mode and summing the counts across all their logins in a semester. The feature value varies drastically across the students in this dataset. For instance, *Number of entries to Guided Study* has a mean of 125.96 and a standard deviation of 116. In contrast, *Time in*

Guided Study has a mean of 18.86 hours and a standard deviation of 15.25 hours per semester. In addition, we calculated normalized features measuring number of hints, number of virtual prizes purchased, and problem accuracies. In total, we mined 110 click-stream features.

2.6 Statistical Analysis

Prior to analysis, all numeric scores were standardized. We used linear mixed effects (LME) models in R [35] using the *lme4* package [7] to develop models of math scores over time (i.e., across the fall and spring semesters). We opted to use LME models because they offer statistical advantages over traditional repeated measures analyses of variance (RM ANOVAs). Specifically, LMEs account for both pooled and individual variance among participants as opposed to only pooled group variance by including subjects as random effects (i.e., assigning a unique intercept for each participant), resulting in more accurate estimates based on individual participant variation. The purpose of the model was to test whether any of the independent variables (e.g., grade level, linguistics and affect features, and click-stream variables) significantly predicted math success. Accordingly, in the model, we entered math success as the dependent variable, with grade level, gender, linguistics and affect features, and click-stream variables as fixed effects (i.e., predictor variables). No interactions were conducted between fixed factors. The baseline grade level was second grade. Grade levels were balanced at around 250–300 students in grades 3–5. There were fewer students in first grade (∼150) and sixth grade (∼15).

To help prevent over-fitting, we removed several variables prior to analysis. First, we conducted correlations between the dependent variables and the independent variables. Any independent variable that did not demonstrate at least a small relationship with the dependent variable ($r \geq 0.100$) was removed from the analysis. Next, we checked for multicollinearity between the remaining independent variables using variance inflation factors (VIF) with a threshold set to 5 (i.e., high multicollinearity). All variables showing VIF above 5 were removed from the analysis and the remaining variables were used in the LME analysis. This variable pruning left us with five click-stream variables ad seven linguistic variables. For each dependent variable, an initial LME model was run with all independent variables. After an initial model was constructed, we used a stepwise variable selection technique (backwards) to eliminate non-significant effects. The results of the stepwise model were used as the final models for the analyses.

We used several other packages to aid in our construction and interpretation of our models. We used *lmerTest* [25] to derive p-values from the models and to perform automatic backward elimination of variables in the LME models, and the *MuMIn* package [6] to obtain two measures of variance explained: a marginal R^2 measuring the variance explained by the fixed effects only, and a conditional R^2 measuring the variance explained by the fixed and random effects combined.

3 Results

An LME model predicting math success as the dependent variable reported significant main effects for a number of click-stream and linguistic features. In general, the click-stream effects indicate that students that were more successful at level A math problems spent less time in the main page of the system (i.e., the *City Landscape*), entered *Guided Study* more, and purchased more items. The linguistic effects demonstrated that students that were more successful at level A math problems produced more sophisticated language (i.e., words with fewer associations, fewer orthographic neighbors and lower range scores), used a greater

Table 1 LME model predicting math success.

Fixed effect	Estimate	Percent of estimate	Std. Error	t	p
(intercept)	0.086		0.064	1.335	0.182
6th grade	-0.512	0.265	0.213	-2.405	0.016
3rd grade	-0.291	0.151	0.080	-3.654	0.000
5th grade	-0.274	0.142	0.083	-3.299	0.001
4th grade	0.219	0.114	0.080	2.758	0.006
Time in City Landscape	-0.137	0.071	0.023	-5.911	0.000
Number of entries into Guided Study	0.094	0.049	0.025	3.765	0.000
Word associations (USF) CW	-0.068	0.035	0.022	-3.097	0.002
Number of items purchased	0.061	0.031	0.023	2.629	0.009
Word range (SUBTLEXus) AW	-0.060	0.031	0.023	-2.577	0.010
Attested constructions (Magazine)	0.060	0.031	0.024	2.490	0.013
Moving Avg. Type Token Ratio (MATTR)	0.057	0.029	0.022	2.541	0.011
Semantic overlap between sentences (word2vec)	0.052	0.027	0.025	2.080	0.038
Orthographic neighbors (CW)	-0.047	0.024	0.023	-2.019	0.044

diversity of words, used more common syntactic constructions, and produced language that was more cohesive. Time was not a significant predictor. The model reported a marginal R^2 of 0.139 and a conditional R^2 of 0.438. Table 1 displays the estimates, percent of estimate, standard errors, t-values, and p-values for the fixed effects for this model.

4 Discussion

This study builds on previous work that examines links between math success and language production by examining how language features and click-stream variables combine to explain student success in an online tutoring system. Unlike previous studies, the current study included student growth over time as a variable. In addition, the click-stream variables examined in this study were finer-grained than in previous studies, allowing us to better understand how student behaviors within the system help explain math success in conjunction with language features. Overall, our fixed factors explained about 14% of the variance in math scores (marginal variance) while a mix of both fixed and random factors explained about 44% of the variance (conditional variance).

In general, the results indicate that grade level was the strongest predictor of math success such that there was a decline in the percentage of level A problems correctly answered among students in higher grades, although fourth-grade students showed a notable departure from that trend. Post-hoc analyses (not reported here) indicated that trends were not linear with fourth graders performing better than third, fifth and sixth graders. We hypothesize that these results may be indicative of a developmental milestone or a change in curriculum expectations (e.g., with competence generally increasing, but fifth and sixth graders receiving more challenging material), but more research is needed.

Beyond grade level, the next strongest predictors were related to click-stream variables. Specifically, the more time that students spent in the *City Landscape*, the lower they performed on A-level math problems. This is not surprising, as no math learning happens in *City*

Landscape. In fact, spending more time in *City Landscape* may suggest the student has lower persistence because they are constantly trying to switch modes in the tutor. In comparison, the more entries they made to *Guided Study*, which is the main instructional and study mode in *Foundations*, the better they performed. We also observe that high student performance was correlated with more purchases in the shopping mall; this is because the points used to make purchases are earned through better mathematics performance. It may be an interesting area of future work to see if differences in the items students purchase relate to differences in math success.

In terms of language production, more successful students produced words that were more sophisticated. For example, more successful students used words that had fewer associations, were found in fewer texts (i.e., a lower range score), and had fewer orthographic neighbors (i.e., words that are spelled similarly). All of these indices indicate that more successful students had more depth of lexical knowledge. Not only were the words they produced more complex, these students also used a greater variety of words (i.e., lexical diversity) indicating that they had larger productive vocabularies (i.e., breadth of lexical knowledge). Beyond lexical knowledge, more successful students also produced a greater number of verb argument constructions indicating a greater range of syntactic structures and produced language that was more cohesive in terms of semantic similarity between sentences. These data indicate that students who are more successful at solving math problems are more proficient language users, in line with previous findings [10, 11, 12, 13, 16].

It is interesting to note that time was not a significant predictor of math success in our LME model. Thus, there is no evidence of improved performance between the fall and spring semester in terms of A-level problems. This is likely related to the curriculum design, which scaffolds student learning and arranges content to become increasingly difficult as a student masters easier content. In addition, gender was not a significant predictor of math success; girls and boys performed similarly on level A problems.

5 Conclusion

The work presented here provides additional evidence that links language production to math success. In general, there seems to be strong evidence that students who are more successful in math produce language that is more sophisticated in terms of words and more complex in terms of syntax. In addition, more successful math students also produce language that is more cohesive and follows language conventions found in adult language corpora.

The study also finds that student choice of activities is associated with their degree of success. Specifically, this study looked at the virtual locations within RM City, which represent different activities within the Foundations curriculum. We found that students who were struggling were more likely to be switching from one activity to the next (through the *City Landscape* mode). while high performing students were more likely to engage in other types of activities, like making purchases for their *My Space*) and spending more time in modes related to math problems.

While we were able to capture many features of student behavior within the system and student language production, there is of course room for further feature engineering. For example, it may be possible to better capture the variation in student behavior in the system through creating additional temporal features. Linguistically, features related to intention and meaning should be deployed as well to increase our knowledge of how language and math skills interact.

Lastly, of interest is the amount of variance explained by the random factors (i.e., the conditional variance). While the fixed factors explained about 14% of the variance in the math score, the majority of the variance (∼30%) was explained by the random factor of participant. This finding may indicate that much of math success is not in behaviors within the system, grade level, or language production but likely rather resides in the individual differences of students. These results suggest that it may be useful in future research to look into which individual differences (e.g., ELL status, geographic location, ethnicity, socio-economic status) may best explain math success.

References

1 Jamal Abedi and Carol Lord. The Language Factor in Mathematics Tests. *Applied Measurement in Education*, 14(3):219–234, 2001.

2 Thomasenia Lott Adams. Reading mathematics: More than words can say. *The Reading Teacher*, 56(8):786–795, 2003.

3 Mary Alt, Genesis D Arizmendi, and Carole R Beal. The relationship between mathematics and language: Academic implications for children with specific language impairment and English language learners. *Language, speech, and hearing services in schools*, 45(3):220–233, 2014.

4 Nathaniel Anozie and Brian W Junker. Predicting end-of-year accountability assessment scores from monthly student records in an online tutoring system. Technical report, Educational Data Mining: Papers from the AAAI Workshop. Menlo Park, CA: AAAI Press, 2006.

5 David A Balota, Melvin J Yap, Keith A Hutchison, Michael J Cortese, Brett Kessler, Bjorn Loftis, James H Neely, Douglas L Nelson, Greg B Simpson, and Rebecca Treiman. The English lexicon project. *Behavior research methods*, 39(3):445–459, 2007.

6 Kamil Barton. *MuMIn: Multi-Model Inference*, 2018. URL: `https://CRAN.R-project.org/package=MuMIn`.

7 Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

8 Carole R Beal, Rena Walles, Ivon Arroyo, and Beverly P Woolf. On-line tutoring for math achievement testing: A controlled evaluation. *Journal of Interactive Online Learning*, 6(1):43–55, 2007.

9 Marc Brysbaert and Boris New. Subtlexus: American word frequencies. `http://subtlexus.lexique.org`, 2009.

10 Scott Crossley, Tiffany Barnes, Collin Lynch, and Danielle S McNamara. Linking Language to Math Success in an On-Line Course. In *Proceedings of the 10th International Conference on Educational Data Mining*, pages 180–185, Wuhan, China, 2017.

11 Scott Crossley and Victor Kostyuk. Letting the Genie out of the Lamp: Using Natural Language Processing tools to predict math performance. In *International Conference on Language, Data and Knowledge*, pages 330–342. Springer, 2017.

12 Scott Crossley, Ran Liu, and Danielle McNamara. Predicting math performance using natural language processing tools. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 339–347. ACM, 2017.

13 Scott Crossley, Jaclyn Ocumpaugh, Matthew Labrum, Franklin Bradfield, Mihai Dascalu, and Ryan S Baker. Modeling Math Identity and Math Success through Sentiment Analysis and Linguistic Features. In *International Educational Data Mining Society*. ERIC, 2018.

14 Scott A. Crossley, Kristopher Kyle, and Mihai Dascalu. The Tool for the Automatic Analysis of Cohesion 2.0: Integrating semantic similarity and text overlap. *Behavior Research Methods*, 51(1):14–27, 2019.

15 Scott A Crossley, Kristopher Kyle, and Danielle S McNamara. Sentiment Analysis and Social Cognition Engine (SEANCE): An automatic tool for sentiment, social cognition, and social-order analysis. *Behavior research methods*, 49(3):803–821, 2017.

16 Scott A Crossley, Maria-Dorinela Sirbu, Mihai Dascalu, Tiffany Barnes, Collin F Lynch, and Danielle S McNamara. Modeling Math Success Using Cohesion Network Analysis. In *International Conference on Artificial Intelligence in Education*, pages 63–67. Springer, 2018.

17 Mark Davies. The 385+ million word Corpus of Contemporary American English (1990–2008+): Design, architecture, and linguistic insights. *International journal of corpus linguistics*, 14(2):159–190, 2009.

18 Mingyu Feng, Neil T Heffernan, and Kenneth R Koedinger. Predicting state test scores better with intelligent tutoring systems: developing metrics to measure assistance required. In *International conference on intelligent tutoring systems*, pages 31–40. Springer, 2006.

19 Pier Luigi Ferrari. Mathematical Language and Advanced Mathematics Learning. *International Group for the Psychology of Mathematics Education*, 2004.

20 Gillian Hampden-Thompson, Gail Mulligan, Akemi Kinukawa, and Tamara Halle. Mathematics Achievement of Language-Minority Students During the Elementary Years. Research report, The University of York, Washington, DC, 2008.

21 Neil T Heffernan, Kenneth R Koedinger, Brian W Junker, and Steven Ritter. Using Web-based cognitive assessment systems for predicting student performance on state exams. *Research proposal to the Institute of Educational Statistics, US Department of Education. Department of Computer Science at Worcester Polytechnic Institute, Massachusetts*, 2001.

22 Federico Hernandez. *The Relationship Between Reading and Mathematics Achievement of Middle School Students as Measured by the Texas Assessment of Knowledge and Skills*. PhD thesis, University of Houston, 2013.

23 Arnon Hershkovitz, Ryan Shaun Joazeiro de Baker, Janice Gobert, Michael Wixon, and Michael Sao Pedro. Discovery with models: A case study on carelessness in computer-based science inquiry. *American Behavioral Scientist*, 57(10):1480–1499, 2013.

24 George A Khachatryan, Andrey V Romashov, Alexander R Khachatryan, Steven J Gaudino, Julia M Khachatryan, Konstantin R Guarian, and Nataliya V Yufa. Reasoning Mind Genie 2: An intelligent tutoring system as a vehicle for international transfer of instructional methods in mathematics. *International Journal of Artificial Intelligence in Education*, 24(3):333–382, 2014.

25 Alexandra Kuznetsova, Per B Brockhoff, and Rune Haubo Bojesen Christensen. lmerTest package: tests in linear mixed effects models. *Journal of Statistical Software*, 82(13):1–26, 2017.

26 Kristopher Kyle, Scott Crossley, and Cynthia Berger. The tool for the automatic analysis of lexical sophistication (TAALES): version 2.0. *Behavior research methods*, 50(3):1030–1046, 2018.

27 Kristopher Kyle and Scott A Crossley. Measuring Syntactic Complexity in L2 Writing Using Fine-Grained Clausal and Phrasal Indices. *The Modern Language Journal*, 102(2):333–349, 2018.

28 Jo-Anne LeFevre, Lisa Fast, Sheri-Lynn Skwarchuk, Brenda L Smith-Chant, Jeffrey Bisanz, Deepthi Kamawar, and Marcie Penner-Wilger. Pathways to mathematics: Longitudinal predictors of performance. *Child development*, 81(6):1753–1767, 2010.

29 Mollie MacGregor and Elizabeth Price. An exploration of aspects of language proficiency and algebra learning. *Journal for Research in Mathematics Education*, 30:449–467, 1999.

30 Maria Martiniello. Language and the performance of English-language learners in math word problems. *Harvard Educational Review*, 78(2):333–368, 2008.

31 Maria Martiniello. Linguistic complexity, schematic representations, and differential item functioning for English language learners in math tests. *Educational assessment*, 14(3-4):160–179, 2009.

32 William L Miller, Ryan S Baker, Matthew J Labrum, Karen Petsche, Yu-Han Liu, and Angela Z Wagner. Automated detection of proactive remediation by teachers in Reasoning Mind classrooms. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge*, pages 290–294. ACM, 2015.

33 Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. The University of South Florida word association, rhyme, and word fragment norms, 1998.

34 Jaclyn Ocumpaugh, Maria Ofelia San Pedro, Huei-yi Lai, Ryan S Baker, and Fred Borgen. Middle school engagement with mathematics software and later interest and self-efficacy for STEM careers. *Journal of Science Education and Technology*, 25(6):877–887, 2016.

35 R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL: `https://www.R-project.org/`.

36 Steven Ritter, Ambarish Joshi, Stephen Fancsali, and Tristan Nixon. Predicting standardized test scores from Cognitive Tutor interactions. In *Proceedings of the International Conference on Educational Data Mining*, 2013.

37 Maria Ofelia San Pedro, Jaclyn Ocumpaugh, Ryan S Baker, and Neil T Heffernan. Predicting STEM and Non-STEM College Major Enrollment from Middle School Interaction with Mathematics Educational Software. In *Proceedings of the International Conference on Educational Data Mining*, pages 276–279, 2014.

38 Maria Ofelia Clarissa Z San Pedro, Ryan SJ d Baker, and Ma Mercedes T Rodrigo. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics. In *International Conference on Artificial Intelligence in Education*, pages 304–311. Springer, 2011.

39 Maria Ofelia Z San Pedro, Ryan SJ d Baker, and Ma Mercedes T Rodrigo. Carelessness and affect in an intelligent tutoring system for mathematics. *International Journal of Artificial Intelligence in Education*, 24(2):189–210, 2014.

40 David Tall. Thinking Through Three Worlds of Mathematics. *International Group for the Psychology of Mathematics Education*, 2004.

41 Rose K Vukovic and Nonie K Lesaux. The relationship between linguistic skills and arithmetic knowledge. *Learning and Individual Differences*, 23:87–91, 2013.

42 Jun Xie, Alfred Essa, Shirin Mojarad, Ryan S Baker, Keith Shubeck, and Xiangen Hu. Student learning strategies and behaviors to predict success in an online adaptive mathematics tutoring system. In *Proceedings of the International Conference on Educational Data Mining*, pages 460–465, 2017.

Can Computational Meta-Documentary Linguistics Provide for Accountability and Offer an Alternative to "Reproducibility" in Linguistics?

Tobias Weber
Institut für Finnougristik, LMU Munich, Germany
http://kraasna.wordpress.com
weber.tobias@campus.lmu.de

Abstract

As an answer to the need for accountability in linguistics, computational methodology and big data approaches offer an interesting perspective to the field of meta-documentary linguistics. The focus of this paper lies on the scientific process of citing published data and the insights this gives to the workings of a discipline. The proposed methodology shall aid to bring out the narratives of linguistic research within the literature. This can be seen as an alternative, philological approach to documentary linguistics.

2012 ACM Subject Classification Applied computing; Applied computing → Anthropology; Applied computing → Publishing

Keywords and phrases Language Documentation, meta-documentary Linguistics, Citation, Methodology, Digital Humanities, Philology, Intertextuality

Digital Object Identifier 10.4230/OASIcs.LDK.2019.26

Category Extended Abstract

1 Introduction

In this position paper, I will propose a methodology which draws from approaches in computational linguistics to help with the goals of meta-documentary linguistics [2]. In a broad definition, this field investigates all processes around a documentary project, including value-adding steps after the recording of data, and tracking metadata. I propose including a new layer to meta-documentations consisting of the continuous tracking of citations and instances where the outputs are used beyond the publication of results - an extended, proactive meta-documentation. The aim of the methodology proposed here is to enable this approach and, subsequently, increase transparency in linguistic research by utilising the recent advances of big data and applying them to a specific element of linguistic publications deriving from a documentary project: the examples.

1.1 Background

Accountability has been at the centre of linguistic research, as in every scientific discipline which draws heavily from primary data. Thus, both "explicit concern for accountability" and "focus on primary data" are essential features for documentary linguistics [9]. Subsequently, accountability in language documentation is facilitated by providing a sufficient amount of metadata to accompany the set of primary data, including information on the technical side of the file or recording, the content, its producers, and the circumstances of its creation (e.g. place and time of recording, type of elicitation, use of stimuli) [13]. It should furthermore highlight all value-added procedures by members of the scientific community, such as transcribing and annotating [14]. However, it is questionable whether any amount of metadata will ever be able to fully convey the narrative behind the the documentation process, with all its


© Tobias Weber;
licensed under Creative Commons License CC-BY
2nd Conference on Language, Data and Knowledge (LDK 2019).
Editors: Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski; Article No. 26; pp. 26:1–26:8

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

details and peculiarities, no matter how well the corpus of language data is "mediated" [10]. To solve this issue, a possible step is to supply a meta-documentation, a narrative about the documentary work and its outputs based on the metadata [24] [2]. At the same time, this opens up the chance to examine the workings of linguistics from the perspective of the History of Scholarship, and interpret the narratives with an anthropological view to highlight the backgrounds and the human factors within such an endeavour [6]. This, in turn, would create the necessary transparency of the documentation project which provides the basis for accountability of all works drawing from its outputs.

As introduced above, meta-documentary linguistics can provide insights to the narratives behind a documentation project and its outputs. Those are usually stored in archives and can be used by researchers for their linguistic work [9]. The most common way of using archived materials is by citing them as examples within a piece of writing. However, citation objectifies language and turns the example sentences into the object of language description - and artefact of the linguistic process. The example is detached from its original context within a story, an archive, or the entire documentation project. It becomes a new instance of this example, a different "version" or "generation", or, using the idea of Basalla's genealogy of artefacts [3], a new artefact mediated by the new technological means with a different interpretative backgrounds.

The aspect of new technological means in the mediation of the linguistic example deserves some attention, as current publishing practices, even electronic ones, are still tied to the paper publication format or a digital replication thereof (e.g. a PDF file), while new formats of publication are only slowly adopted [21]. This also means that any metadata attached to the original set of primary data, for example in an XML file [1], might be deleted due to restrictions in the medium or overwritten by new metadata of the citation, i.e. contain traces from the new technology [19]. However, the focus of attention here is the different interpretative background, as data which is frequently cited will accrue a variety of contexts and descriptive/theoretical frameworks in which it is used over the course of time. It presents us with the issue of intertextuality.

Frequent use of the same example might lead to changes in the representation of the data, interpretations derived from them, and features attributed to them. This is the point where linguists assume the agency to manipulate the data to fit into their theoretical frameworks according to their own convictions and beliefs about a particular language, grammatical feature, or the nature of language itself. While the term "manipulate" sounds more severe than the action it describes, changing the grammatical tag on a morpheme, editing morpheme boundaries, or adding a new translation (all value-adding processes [2]) can lead to entirely different interpretations of the language and its grammar. And those processes of adding value are routinely executed by linguists as part of their profession and without any mischievous intent. However, this can distort the provenance of a particular interpretation or presentation to an extent where the author themselves cannot determine where an interpretation came from; the scientific community engages with the data and with other members within the community, the moment of generating new ideas and thoughts. Yet, this also means that the trajectory of thought cannot easily be traced and that important information on the narrative (or rather: for the meta-documentation) becomes obscured.

1.2 Motivation

In early 2018, a position paper by 14 linguists and 41 undersigned members of the scientific community directed broad attention to this issue, calling for "reproducibility" in linguistic research as a means to create "verification and accountability" [4]. The authors consider

"proper" data citation and attribution as the basis of scientific reproducibility, stressing the central role of primary data within the field. And while it cannot be argued that correct and transparent handling of data is the tenet of our discipline and should be fostered and enhanced, the notion of "reproducibility" is firmly based in empirical, quantitative sciences where data are reified and purely objective (this kind of data "speak for themselves"). This hints at an understanding of linguistics where conventions and the professional conduct of the researchers as objective instances guarantee identical procedures in handling data which leads to reproducible results - and it ignores all instances of researchers assuming the agency to "break" rules, come up with innovative approaches, or add value to the data. To ensure brevity of this paper, I will not discuss the position paper in further detail but move on by pointing out the crucial element which is missing in this approach: the human factor and the intertextuality.

As initially stated, a central aspect to any set of metadata is the meta-documentation which is elaborating on the data set and providing insights to the contexts of creation of the data. This concept provides a narrative of the documentary work and creates transparency about it and, most importantly, the instances of human interaction with the data. It is, thus, an opposite idea to the objective stance of "reproducibility", where the goal is to reduce the human factor in our work. And even though both approaches are tied to the original data set and require the researcher to access and make reference to the archived originals, they cannot prevent that the researcher interacts with the data in creative ways. To ensure that information on these interactions is not lost, tracking of the citation of examples should be used, either as metadata or as a meta-documentation which proactively anticipates such interactions and can expand accordingly. In my opinion, the meta-documentary approach seems more sensible, as it does not restrict the researchers' agency in working with the data and provides a basis to document and record the human factor, and the subjectivity, of linguistic research instead of imposing a rigorous objective stance.

An alternative, humanistic approach is chosen by Frank Seidel [17]. He describes documentary linguistics as a "philology" which pays attention to the contexts of data and allows for variation instead of rigour. However, the philological approach can not only be applied to primary data but should be extended to also cover all instances of data citation in the literature. It emphasises the role of the researchers and all decision makers within the publication process (e.g. editors of journals, archivists, the scientific community) and provides information relevant for tracing the trajectory of thought and commenting on variation.

Focusing on the citation of language examples, there are many instances where variation can be observed. A great survey of the "defective documentation" of Norwegian in published linguistics literature can be found in the work by Jan Engh [7]. In this survey, Engh examines mistakes in data citation for Norwegian, which, in comparison to many endangered languages, provides two literary standards with national status and various resources and reference materials available. As such, it illustrates well how the researcher's hand can influence the outcomes of citing a language example, and provides a strong counter-argument to the calls for more objective research. A second interesting point to be found in Engh's work is that, in some instances, an error was inherited from a secondary source. I would therefore like to conclude that tracking citations of language examples is necessary for a thorough meta-documentation: it should cover all instances of citation, their contexts, and their relationships to other instances. Other examples for different languages could easily be named.

In my own work [23], I have worked on legacy materials of the extinct South Estonian Kraasna dialect, building a meta-documentation and tracing the use of examples through the literature. This produced a list of differences between original transcriptions and published materials, as well as between secondary sources. While, in the case of Kraasna, the bulk of data to be handled was considerably small, frequently cited languages and documentation projects pose a different obstacle for any researcher trying to trace examples and restore a missing meta-documentation manually. However, using a mixed methods approach [11], could save a researcher from tedious work while providing a solid basis for the preparation of a meta-documentation, and furthermore enabling research into scientific practices which are invisible to the human eye but clear to the computer.

Ultimately, I would like to argue that focusing on language examples cannot only help to handle the variation in representation but also share insights on the interpretative contexts for their citation. As (false) versions can be inherited from the secondary literature, interpretations can also be inherited or at least shaped by previous analyses on an example. I would even argue that, in an instance where a previous interpretation is completely negated, there is still an inherited element - the acceptance of an example as representative for a language, of an author within the scientific community, or a piece of writing within the canon of scientific works on a topic. The same holds true for all instances where an example is identical in its version but not in its citation context. The only way to ensure that the original metadata of the "artefact" are not altered is to completely ignore and exclude the example from scientific procedures.

2 Proposed Methodology

In the previous section, I presented the ideas behind meta-documentary linguistics and how they can help linguistic research to become more transparent and accountable. I, subsequently, propose the application of computational methods to the tracking of language examples within linguistic literature and the resulting computational meta-documentary linguistic research as a combination of humanistic/philological strands of documentary linguistics and computational linguistics. Although neither the computational tools and methods in question, nor the use of automatic citation trackers are novel, the application to language examples as parts of publications is nowadays easier than before. A first approach to compiling language examples can be seen in ODIN [12], however this project appears to be resting since 2010.

2.1 Goal

The goal of the application would be to search and establish relationships (a "genealogy") of linguistic examples within (a defined set of) linguistic publications. This could help to show relations between works even if they are not indicated in the publication, and would also provide a continuous trajectory of the citations [8]. Such a tool could be useful for archivists, documenters, the scientific community, as well as publishing houses. It would provide a valuable addition to the metadata of a file, a self-updating meta-documentation beyond the initial compilation, a "database-like" platform for linguistic examples, and might integrate well with already existing anti-plagiarism software [20]. For the collectors of the data, this tool might help to check the trajectory of the own work and its distribution. Furthermore, it might uncover biases in the citation of a particular language or particular example, e.g. by a certain schools, frameworks, or researchers.

2.2 Workflow

In order to achieve these goals, the application needs to

- search literature and identify linguistic examples;
- read data from the literature and store it;
- compare variants and classify them accordingly;
- compare with cited or indicated sources / the references of the source (optional); and
- collate the results in a presentational format.

Searching

The task of searching literature might be abridged by providing a preexisting database of publications, supplying relevant URLs, or a set of PDF files [25]. However, it is also possible to think of a crawler, which could search for relevant sources online. This function needs to detect linguistic data from plain text, OCRed text, and would, ideally, be able to apply OCR techniques to images. Luckily, linguistic examples are usually presented in a conventionalised format [5] which singles out examples to a separate block in the text, generally assigning a number and providing information on metadata (e.g. language name, source, date, reference). In addition, from a plain text or a suitable OCRed version of an image, the application might receive information about the character set used, which can be an indicator in the case of a phonetic transcription like the International Phonetic Alphabet. Within running text, examples might be italicised and followed by an analysis or translation.

Reading

The step of reading and storing the data requires a high amount of storage space, or a smart search algorithm which allows to reduce the stack for comparison tasks. For this step, methods used in big data applications or anti-plagiarism software could be used. The stored examples would need to be normalised to the extent that they are comparable but not additionally altered by the software; the originally intended rendering must be preserved, or at least stored with the normalised version. It is obvious that this function needs to support Unicode with all phonetic characters and diacritics.

Comparing

The comparison task would utilise common functions of pattern matching or fuzzy string searching [15] to establish the differences between versions and group them accordingly. Should the application support the optional functionality of including metadata on the potential source found within the publication, or other relevant information like the name of the author(s) or the year of publishing, these pieces of information would have to be collected in the first step and similarly matched in the comparing step [8].

Collating

The final step would be to collate the information gathered about each group of examples and arranging them by using a hierarchy of filters, with time and type of version being the primary criteria. Drawing from philology, such a textual genealogy (in Lachmannian tradition) can be translated into a path graph with the root node being the archived original transcription and each version being attributed to a parent node. Presumably, the results of this process will contain some unclear relations which would need to be resolved; the solution to this issue, however, requires theoretical considerations to be made about the required certainty for establishing a relation, which is not primarily a technical concern.

2.3 Format of the Output

As indicated above, the result of the procedure would be a list of clustered/grouped versions, or a path graph using time as an ordering principle. The presentation of the results would depend on the nature of the query, whether a particular version is searched within the database, all versions of a single original shall be displayed, or a bulk query is made about a particular project, author, or language. Potential presentational formats range from knowledge graphs in the style of the Web of Science [22], simple tree graphs, or lists and other plain text formats (e.g. CSV). However, the functionality of the application must not depend on the format of the output, which is rather a concern for the design of an interface or front-end to this tool and, therefore, highly dependent on the integration into other digital infrastructures.

3 Potential Obstacles

Before concluding this discussion, I would like to highlight some potential issues and pitfalls of the proposed methodology. Firstly, for collecting examples from the literature, access and usage rights would have to be negotiated with the copyright holders or the publishing houses. Although there are several open access journals in the field of linguistics [16], most high-profile journals and publications are still requiring the acquisition of access rights for their articles behind a paywall. The same holds true for various archives, where access rights are limited. It might be possible to agree a collaboration with publishing houses or a library to gain access for the crawler, however, the only way forward for transparency in published research seems to be open science.

Secondly, the type of the crawled file can influence the results greatly. No matter how good OCR systems have become, there are still issues with older prints, uncommon fonts, or with text donning an array of different, yet optically very similar, diacritics (like a breve and a hačekˇ). This becomes even more difficult as citations in the running text have to be recognised and not mistaken for normal text, which can make the delimiting of the example very difficult.

Thirdly, there are various types of transcription used, which also have to be attributed to the correct original source. For example, languages using a special writing system different from the Latin alphabet (e.g. various Asian languages, languages in Russia) might be transliterated to a Latin-based version; the solution would have to be a built-in transcription tool or optional integration for such a tool from external sources which use standardised transliteration (like ISO 9 for Cyrillic). Additionally, there are subfields of linguistics which are actively using their own, traditional transcription systems like the UPA for Uralic linguistics [18]. And even though IPA is already advancing in those fields, the proposed tool should be able to handle earlier sources equally well.

Fourthly, the sheer amount of data which needs to be crawled and the required computing power will mean that this application cannot be run every time a query is made by the user, or rather that there needs to be a good balance between results which are stored in a database file (for example in a CSV format) and queries requiring a new searching process. As both, storage space and computing power would be excessively used by the task, a feasible and efficient solution for the storage and presentation of the results would have to be found.

Lastly, the question remains whether queries are sent as bulk or for specific example sentences. Should the user have the chance to enter a sentence to be compared to the database, or would there only be access to a preexisting set of examples (e.g. stored with the meta-data for a particular file containing primary data). While it would be desirable

to allow user interaction with the tool and provide a powerful application for public use, it appears that such a methodology would have to be used sparingly until the fourth point can be sufficiently resolved.

4 Outlook

In this paper, I discussed how the application of rather common methods from computational linguistics could help to make linguistic research and the use of primary data within the discipline more transparent. The proposed methodology focuses on the linguistic examples within the published research and could provide important insights to the workings of this field. This methodology should be seen as an example of the increasing use of technology within the humanities, in particular for language documentation and the theorisation thereof – a potential field for future enquiry which might be understood as computational meta-documentary linguistics.

At the moment there are no concrete plans to create an application following the outlined workflow. Yet, this project might yield interesting results about the use of primary data in linguistics, which is an explicit concern for researchers in the field and currently widely discussed among scholars. Whether or not it is possible to conduct the survey in exactly the suggested way will depend on solutions to the obstacles outlined in the previous section, especially with regards to computing power, storage, and access rights. However, as technology advances at a high pace, it will likely be possible to handle, process, and store the necessary amount of data with relative ease in the future - an obstacle which should be overcome in the next decade. In order to obtain access and usage rights from the copyright holders, there could be possible solutions to agree on a strategic partnership with publishing houses or major libraries, to acquire research grants and support by influential scientific interest groups, and to support open access and open science movements. Although this issue is not as easily resolved as others, it can be hoped that necessary agreements will be made and that a successful pilot may convince an increasing amount of rightsholders.

A subsequent idea for increasing transparency in linguistic research regarding the citation of primary data could draw from the results of the application outlined here: If there is a data set on the relations between different versions of a particular example, this could be encoded in a standardised identifier format for linguistic examples, with a full genealogy accessible in a database connected to the identifier. In other words, version control for example sentences.

Finally, I encourage all computational linguists and data scientists to consider how their knowledge and skills might be applied within their own discipline, as a tool to aid metascience from a humanistic point of view.

References

1 Peter K. Austin. Data and language documentation. In Jost Gippert, Nikolaus P. Himmelmann, and Ulrike Mosel, editors, *Essentials of Language Documentation*, pages 87–112. Mouton de Gruyter, Berlin, New York, 2006.

2 Peter K. Austin. Language documentation and meta-documentation. In Mari Jones and Sarah Ogilvie, editors, *Keeping Languages Alive. Documentation, Pedagogy, and Revitalisation*, pages 3–15. Cambridge University Press, 2013.

3 George Basalla. *The Evolution of Technology*. Cambridge University Press, 1988.

4 Andrea L. Berez-Kroeker, Lauren Gawne, Susan Smythe Kung, Barbara F. Kelly, Tyler Heston, Gary Holton, Peter Pulsifer, David I. Beaver, Shobhana Chelliah, Stanley Dubinsky, Richard P. Meier, Nick Thieberger, Keren Rice, and Anthony C. Woodbury. Reproducible research in linguistics: A position statement on data citation and attribution in our field. *Linguistics*, 56(1):1–18, 2018.

5 Max Planck Institute for Evolutionary Anthropology Department of Linguistics. Leipzig Glossing Rules. Conventions for interlinear morpheme-by-morpheme glosses. URL: https://www.eva.mpg.de/lingua/resources/glossing-rules.php.

6 Lise M. Dobrin and Josh Berson. Speakers and language documentation. In Peter K. Austin and Julia Sallabank, editors, *The Cambridge Handbook of Endangered Languages*, pages 187–211. Cambridge University Press, 2011.

7 Jan Engh. *Norwegian examples in international linguistics literature. An inventory of defective documentation.* Universitetsbiblioteket i Oslo, Oslo, 2006.

8 Bela Gipp. *Citation-based Plagiarism Detection.* Springer Vieweg, Wiesbaden, 2014.

9 Nikolaus P. Himmelmann. Language documentation: What is it and what is it good for? In Jost Gippert, Nikolaus P. Himmelmann, and Ulrike Mosel, editors, *Essentials of Language Documentation*, pages 1–30. Mouton de Gruyter, Berlin, New York, 2006.

10 Gary Holton. Mediating language documentation. In David Nathan and Peter K. Austin, editors, *Language Documentation and Description 12: Special Issue on Language Documentation and Archiving*, pages 37–52. SOAS, London, 2014.

11 R. Burke Johnson and Anthony J. Onwuegbuzie. Mixed Methods Research: A Research Paradigm Whose Time Has Come. *Educational Researcher*, 33(7):14–26, 2004.

12 William Lewis. ODIN - The Online Database of Interlinear Text. URL: http://odin.linguistlist.org/.

13 David Nathan. Archives 2.0 for endangered languages: From disk space to MySpace. *International Journal of Humanities and Arts Computing*, 4(1-2):111–124, 2010.

14 David Nathan and Peter K. Austin. Receonceiving metadata: language documentation through thick and thin. *Language Documentation and Description*, 2:179–187, 2004.

15 Gonzalo Navarro, Ricardo Baeza-Yates, Erkki Sutinen, and Jorma Tarhio. Indexing Methods for Approximate String Matching. *IEEE Data Engineering Bulletin*, 24:19–27, 2001.

16 OpenEdition. URL: https://www.openedition.org/.

17 Frank Seidel. Documentary linguistics: A language philology of the 21st century. *Language Documentation and Description*, 13:23–63, 2016.

18 Emil Nestor Setälä. Über die transskription der finnisch-ugrischen sprachen. *Finnisch-ugrische Forschungen*, 1:15–52, 1901.

19 Edward Shils. *Tradition.* The University of Chicago Press, 1981.

20 Benno Stein, Sven Meyer zu Eissen, and Martin Potthast. Strategies for retrieving plagiarized documents. In Wessel Kraaij and Arjen P. de Vries, editors, *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 825–826. ACM, 2007.

21 Nicholas Thieberger. Steps toward a grammar embedded in data. In Patricia Epps and Alexandre Arkhipov, editors, *New Challenges in Typology: Transcending the Borders and Refining the Distinctions*, pages 389–407. Mouton de Gruyter, Berlin, New York, 2009.

22 Web of Science. URL: http://wokinfo.com/.

23 Tobias Weber. Kraasna - A page on the Estonian Kraasna maarahvas and its dialect. URL: https://kraasna.wordpress.com/.

24 Anthony C. Woodbury. Archives and audiences: Toward making endangered language documentations people can read, use, understand, and admire. In David Nathan and Peter K. Austin, editors, *Language Documentation and Description 12: Special Issue on Language Documentation and Archiving*, pages 37–52. SOAS, London, 2014.

25 Jian Wu, Pradeep Teregowda, Juan Pablo Fernández Ramírez, Prasenjit Mitra, Shuyi Zheng, and C. Lee Giles. The Evolution of a Crawling Strategy for an Academic Document Search Engine: Whitelists and Blacklists. In *Proceedings of the 4th Annual ACM Web Science Conference*, WebSci '12, pages 340–343, New York, NY, USA, 2012. ACM. doi:10.1145/2380718.2380762.