Achieving High Quality Knowledge Acquisition using Controlled Natural Language

Tiantian Gao*

Department of Computer Science, Stony Brook University, Stony Brook, NY, USA
tiagao@cs.stonybrook.edu

Abstract

Controlled Natural Languages (CNLs) are efficient languages for knowledge acquisition and reasoning. They are designed as a subset of natural languages with restricted grammar while being highly expressive. CNLs are designed to be automatically translated into logical representations, which can be fed into rule engines for query and reasoning. In this work, we build a knowledge acquisition machine, called KAM, that extends Attempto Controlled English (ACE) and achieves three goals. First, KAM can identify CNL sentences that correspond to the same logical representation but expressed in various syntactical forms. Second, KAM provides a graphical user interface (GUI) that allows users to disambiguate the knowledge acquired from text and incorporates user feedback to improve knowledge acquisition quality. Third, KAM uses a paraconsistent logical framework to encode CNL sentences in order to achieve reasoning in the presence of inconsistent knowledge.

1998 ACM Subject Classification I.2.1 Applications and Expert Systems

Keywords and phrases Logic Programming, Controlled Natural Languages, Knowledge Acquisition

Digital Object Identifier 10.4230/OASIcs.ICLP.2017.13

1 Introduction

Much of human knowledge can be represented as rules and facts, which can be used by rule engines (e.g., XSB [22], Clingo [9], IDP [5]) to conduct formal logical reasoning in order to derive new conclusions, answer questions, or explain the validity of true statements. However, rules and facts extracted from human knowledge can be very complex in the real world. This will demand domain experts to spend a lot of time on understanding the rule systems in order to write logical rules. CNLs emerge as better knowledge acquisition systems over rule systems in that they can acquire knowledge from text and represent the text in logical forms for reasoning. CNLs are designed based on natural languages, but with restricted grammar to avoid ambiguities while being highly expressive. Representative languages include ACE [7], Processable English (PENG) [24], BioQuery-CNL [6]. In general, CNL systems provide a GUI for user to enter CNL text. The language parser checks the grammar of the text and sends back suggestions for correction to the user. CNL text is then mapped into the corresponding logic programs based on the syntax and semantics of the underlying rule engine in order to perform question answering tasks.

Though the aforementioned systems have good intent of design, we found that there are several limitations in current CNL systems. First, they have limited ability to identify sentences that express the same meaning but in various syntactical forms. For instance,

* The author is co-advised by Michael Kifer and Paul Fodor from Stony Brook University.

Technical Communications of the 33rd International Conference on Logic Programming (ICLP 2017).
Editors: Ricardo Rocha, Tran Cao Son, Christopher Mears, and Neda Saeedloei; Article No. 13; pp. 13:1–13:10

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

ACE translates sentences `Mary owns a car` and `Mary is the owner of a car` into two different logical representations. As a result, if the first sentence is entered into the knowledge base, the reasoner will fail to answer the question `who is the owner of a car`. However, in the real world, it is very common that the user writes questions in a different way from the author who composes the knowledge base. Second, current CNL systems do not accept inconsistent knowledge to occur. In other words, once inconsistent information is found, the underlying rule engine will break and not be able to conduct any inference tasks. In our view, inconsistency is very likely to happen when the knowledge base is formed by merging multiple resources together. Hence, it is useful to design a paraconsistent logical framework that can reason in the presence of inconsistent knowledge. Third, there is no way for the user to edit or audit the acquired knowledge. CNL systems are not guaranteed to always return the user-expected results. As a result, it is necessary to provide a mechanism for the user to edit the acquired knowledge as opposed to re-write sentences many times in order to meet the requirement.

In this work, we design a knowledge acquisition system, KAM, that achieves three goals. First, KAM performs deep semantic analysis of English sentences and maps sentences that express the same meaning via different syntactic forms to the same standard logical representations. Second, KAM performs valid logical inferences based on the facts and rules extracted from English sentences and achieves inconsistency-tolerance for query answering. Third, KAM builds an environment to assist users with entering and disambiguating English texts. In the following parts, Section 2 shows the background knowledge of the natural language processing tools KAM uses in the knowledge acquisition process, Section 3 describes the architecture of the system, Section 4 shows the current state of research and discusses some open issues, and Section 5 concludes the paper.

2 Background

In this section, we provide the background of linguistic databases, semantic relation extraction, and word similarity measures in the field of natural language processing in order to help readers understand KAM better.

2.1 Linguistic Databases

KAM uses a lexical database, BabelNet, and a frame-relation database, FrameNet, in the process of knowledge acquisition. A lexical database is a database of words. It contains the information of part-of-speech, word sense, *synset* and semantic relations of words. WordNet [19] is one of the famous linguistic databases, where each word is defined with a list of word senses. Words that share similar meanings are grouped as a synset. Synsets are connected by semantic relations. For instance, the *hypernym* relation says that one synset is a more general concept of the other, i.e., `human` is the hypernym of `homo sapiens`. WordNet is rich in word knowledge, but it does not have enough information about named entities we encounter in every life or some specialized fields. DBPedia [1], WikiData [27], and YAGO [25] are databases of entities, where each is defined with a set of properties and the relations with other entities or with some pre-defined ontological classes. However, there is no link between an entity in an entity database like DBPedia and a concept in WordNet. As a result, there is no way to find the semantic relation between a name entity and a concept in WordNet, which is useful in many cases. BabelNet solves this problem by integrating multiple knowledge bases, including WordNet, DBPedia, Wikidata, etc. Besides, it automatically finds the mapping across different knowledge bases and therefore bridges the gap between concepts and entities.

FrameNet is a database representing entity relations using *frames*. A frame consists of a set of *frame elements* and *lexical units*. A frame element denotes an entity that serves a particular semantic role in a frame relation. Frame elements are frame-specific. Therefore, they are not shared among frames. A lexical unit indicates a target word in a sentence that triggers a frame relation. For example, the sentence `Mary works for IBM as an engineer` semantically entails the `Being_Employed` frame relation, where `work` is the lexical unit and `Mary`, `IBM`, and `engineer` represent the `Employee`, `Employer`, and `Position` frame elements respectively. In FrameNet, each lexical unit is associated with a set of *valence patterns* and *exemplar sentences*. Valence patterns show the *grammatical functions* [13] of each frame element with respect to the lexical unit. For the above sentence, `Mary` is the `external` of `work`, and `IBM` and `engineer` are the `dependent` of the prepositional modifiers of `work`. Exemplar sentences are the sample English sentences that realize the valence patterns.

In addition to FrameNet, VerbNet [23] and PropBank [14] are also databases of entity relations. VerbNet and PropBank are purely verb-oriented. Therefore, they cannot recognize noun-, adjective-, or adverb-triggered relations. Besides, since VerbNet and PropBank group verbs based on the syntactic patterns of verbs with respect to the entities, verbs that belong to the same class may not represent the same meaning. The advantage of VerbNet over FrameNet is that VerbNet assigns a WordNet synset ID to each verb. Additionally, it defines an ontology that defines the semantic restrictions for entities that can serve particular semantic roles in an entity relation. In KAM, we use FrameNet augmented with BabelNet synset IDs for each frame element and lexical unit.

2.2 Semantic Relation Extraction

Semantic relation extraction tools analyze the semantics of English sentences and extract their entailed relations. Representative tools include Ollie [18], Stanford Relation Extractor [26], LCC [16], SEMAFOR [4], and LTH [12]. Ollie is a relation extractor that extracts triples representing binary relations based on open domains. Stanford Relation Extractor and LCC, on the other hand, can only extract from a fixed set of relations. Although Ollie is flexible at extracting relations, it cannot standardize triples that represent the same semantic relation. Stanford Relation Extractor and LCC are better at relation standardization, but can work with a limited number of relations.

Compared with the aforementioned tools, SEMAFOR and LTH are FrameNet-based semantic parsers that aim to identify a large number of relations and achieve standardization. Basically, they use machine learning algorithms to train the model based on the exemplar sentences in FrameNet. Based our empirical study, SEMAFOR and LTH do not perform well enough for knowledge acquisition. Recall the sentence `Mary works for IBM as an engineer` from the previous section. SEMAFOR extracts two frames: one is `usefulness` frame triggered by `work`, where `Mary` and `for IBM` represent the `entity` and `purpose` frame elements respectively; the other one is `People_by_vocation` frame triggered by `engineer`, with no frame elements attached. The first one is wrong because `for` in this context does not express the purpose meaning. Although the second frame is correct, it does not find who holds this vocation.

In our analysis of FrameNet 1.6 data, 70.2% valence patterns have only one exemplar sentence and 12.8% valence patterns have two exemplar sentences. However, there are also valence patterns with more than 100 exemplar sentences. An uneven distribution of the exemplar sentences per valence pattern will result in an imprecise estimation of model parameters. In addition, frame elements do not have semantic restrictions, which are useful in practical cases. For instance, comparing sentences `Mary has a full-time job` and `Mary`

`has a well-paid job`, both `full-time` and `well-paid` are adjective modifiers of `job`. But, they are classified as two different frame elements: `Contract-basis` and `Compensation` respectively. Without any semantic constraints, we cannot distinguish these two frame elements based on their syntactical context.

2.3 Semantic Similarity

Semantic similarity measures the semantic closeness between a pair of synsets. In general, there are three classes of methods to compute semantic scores. The first class measures the text similarity of the glosses of between two synsets, where a gloss refers to the English description of the meaning of a word. The representative method includes Lesk [2], where the semantic score is calculated based on the degree of overlapping information between their glosses. The second class measures the distance of the synsets in WordNet. In WordNet, a synset is connected by some semantic edges (i.e., hypernym, hyponym). A simple and intuitively way to measure the semantic similarity is to compute the shortest path between two synsets in WordNet. Therefore, synsets with shorter path lengths have stronger semantic connections. Representative methods include wup [28], lch [15], jcn [11], lin [17], res [21], hso [10]. Recently, with the advancement of machine learning, we can represent a synset by a vector of arbitrary dimensions, where the synset vector is obtained by training a large set of corpus. The representative method includes NASARI [3]. Based on the vector representations of synsets, we can measure the semantic similarity by computing the cosine similarity, weighted overlaps [20] of the vectors. In KAM, we use the NASARI approach. For one thing, NASARI dataset is based on BabelNet, which is more up-to-date than WordNet. Second, NASARI approach shows better performance based on the experimental results shown in [3].

3 KAM Framework

KAM consists of two parts: supervised knowledge annotation and knowledge acquisition. Supervised knowledge annotation is designed to create a Prolog knowledge base that represents an augmented version of FrameNet data. Basically, the knowledge base includes the logical representations of frames, frame elements, lexical units, and valence patterns. Besides, each frame element is assigned with a list of BabelNet synsets that capture its definition. Users can also add new frames to the knowledge base. The Prolog knowledge base is used in knowledge acquisition, where we provide a tool that achieves the following:

1. run deep semantic analysis of controlled English text in order to ensure that different sentences that express the same meaning are mapped to the same logical representations.
2. perform valid logical inferences based on the facts and rules extracted from English sentences and achieve inconsistency-tolerance in the process of knowledge acquisition.
3. allow the user to enter controlled English text, disambiguate acquired knowledge, and perform question answering tasks

3.1 Preliminaries

First, we give a brief overview of KAM's language parser, Attempto Parsing Engine (APE), which is based on ACE grammar[1]. APE translates CNL sentences into a Discourse Representation Structure (DRS)[2], which captures the semantic meaning of the sentences. A

[1] `http://attempto.ifi.uzh.ch/site/docs/syntax_report.html`
[2] `http://attempto.ifi.uzh.ch/site/pubs/papers/drs_report_66.pdf`

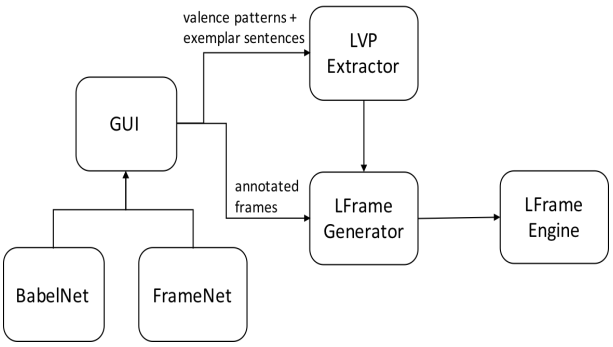

Figure 1 Supervised Knowledge Annotation.

DRS uses six pre-defined predicates to represent the semantics of a word in a sentence, including `object`, `property`, `relation`, `modifier_adv`, `modifier_pp`, `has_part`, `query`, and `predicate` predicates. For instance, the sentence `A man enters a door with a card` is represented as

```
object(A,man,countable,na,eq,1)
object(B,door,countable,na,eq,1)
object(C,card,countable,na,eq,1)
predicate(D,enter,A,B)
modifier_pp(D,with,C)
```

where the `object`-predicate denotes the head word of a noun phrase, the `predicate`-predicate represents an action, and the `modifier_pp` signifies a prepositional modifier to the action.

We define the semantic relation between two predicates as a *dependency path* that connects these two predicates via a list of variables and intermediate predicates. For the above example, `man` is the subject of the `enter` action. The semantic relation is represented as

```
predicate(D,enter,A,B) -> A -> object(A,man,countable,na,eq,1)
```

There can be more than one dependency paths that connect two predicates. For the rest of this section, we will only consider the shortest dependency path.

3.2 Supervised Knowledge Annotation

Figure 1 shows the architecture of supervised knowledge annotation. The GUI provides an environment for the user to annotate FrameNet frames and query BabelNet. Given a frame, the user is required to disambiguate each frame element name by assigning a BabelNet synset to it. For instance, in `Being_Employed` frame, `Position` is assigned with the synset `bn:00010073n` (a job in an organization) and `Employee` is assigned with the synset `bn:00030618n` (a person who is hired to perform a job). The annotated frame and frame elements are mapped into Prolog representation by LFrame Generator as

```
frame_def(Frame_Name,[
    frame_element(Frame_Element_Name, BabelNet_SID)|...])
```

Next, the user annotates each lexical unit and its exemplar sentences. Given that FrameNet exemplar sentences are written in normal English, some may not be parsed by APE. Therefore, the user needs to manually rephrase each exemplar sentence according to ACE grammar.

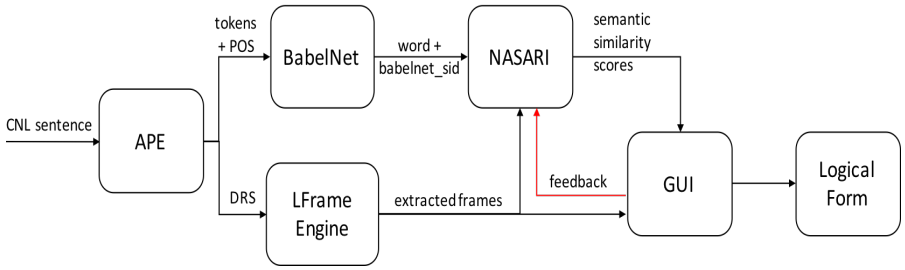

Figure 2 Knowledge Acquisition.

Besides, the user marks the lexical unit and frame elements of a sentence. The annotated lexical units and exemplar sentences are mapped into Prolog representation by LVP Extractor as

```
lvp(Lexical_Unit, Frame_Name, [
    lgf(Frame_Element_1,Dependency_Path_1)|...])
```

where it extracts dependency paths that represent the semantic relations between a lexical unit and the frame elements.

LFrame Engine uses the `frame_def` and `lvp` predicates to extract frame relations and identify frame elements from CNL sentences. Specifically, LFrame Engine applies the `lvp` to each word of a sentence to extract potential frames and frame elements, denoted as

```
frame(Frame_Name,[
    frame_element(Frame_Element_Name, Val)|...])
```

3.3 Knowledge Acquisition

Figure 2 shows the process of translating a CNL sentence into its logical form. First, APE parses the input sentence and generates the DRS and part-of-speech of each word. Second, KAM queries BabelNet and gets the synsets each word belongs to. In parallel, LFrame Engine extracts the candidate frames and frame elements from the DRS.

Next, for each candidate frame relation, KAM disambiguates the word sense of each frame element based on the frame element name. Recall from the previous subsection, each frame element name is assigned with a BabelNet synset ID that captures its definition. Here, KAM uses NASARI database to measure the semantic similarity between each synset the frame element belongs to and the frame element name. KAM chooses the synset with the highest semantic similarity score as the word sense of the frame element. The sum of the semantic scores of each frame element is defined as the score of the extracted frame relation. Finally, KAM ranks the candidate frames based on their scores. For example, given the sentence `There is a person who works in London`, LFrame Engine finds three candidate frame relations:

```
frame(Being_Employed,[frame_element(Employee, person),
    frame_element(Employer, London)])
frame(Being_Employed,[frame_element(Employee, person),
    frame_element(Position, London)])
frame(Being_Employed,[frame_element(Employee, person),
    frame_element(Place, London)])
```

KAM computes the semantic similarity scores between `person` and `Employee` (resp. `London` and `Employer`, `London` and `Position`, and `London` and `Place`) in order to disambiguate the word sense of `person` and `London` in each frame. In this case, the third frame has the highest score where person is assigned with BabelNet synset `bn:00046516n` (a human being) and London is assigned with `bn:00013179n` (the capital and largest city of England). KAM shows the ranked results to the user and asks the user to choose the one which is consistent with his/her understanding. Given that NASARI uses a statical approach to measure the semantic similarities, there could be errors in the computation. KAM allows the user to audit the result. The feedback will be recorded in order to improve the quality of semantic similarity measures in the next run.

3.4 Logical Representation

KAM represents the semantics of the frame relations in a paraconsistent logical framework, Annotated Predicate Calculus (APC) [8]. APC is a paraconsistent logical framework that deals with inconsistency. The syntax is the same as FOL except for atomic formulas of the form $p : s$, where p is an FOL atomic formula and s is a truth annotation. Truth annotations come from an arbitrary upper the Belnap's semilattice with four truth values: $\bot$, $\mathbf{t}$, $\mathbf{f}$, $\top$ where $\bot \leq \mathbf{f} \leq \top$ and $\bot \leq \mathbf{t} \leq \top$. Here, $\mathbf{t}$ and $\mathbf{f}$ denote a predicate is true and false respectively. $\bot$ denotes a predicate is neither true or false. $\top$ denotes a predicate is both true and false, which causes an inconsistency. APC is based on stable model semantics and the models are computed on Clingo. Further details of APC and its applications in natural language understanding can be found in [8]. The advantage APC provides over Answer Set Programming (ASP) systems and first-order logic is that APC allows inference in the presence of inconsistent knowledge. Besides, it captures a lot of complex features in natural language, e.g., negation, numerical constraints, reasoning by cases. For the previous sentence `There is a person who works in London`, its encoding is

```
frame(being_employed, #1) : t.
frame_element(#1, employee, #2) : t.
frame_element(#1, place, #3) : t.
object(#2, person, bn:00046516n) : t.
object(#3, london, bn:00013179n) : t.
```

where `t` is a truth annotation in APC, `#1`, `#2`, and `#3` are skolemized constants, `bn:00046516n` and `bn:00013179n` refer to BabelNet synsets.

4 Evaluation Design

Our initial step of evaluation is to test CNL sentences which describe human-related information, including a person's gender, occupation, origin, age, nationality, religious belief, and so on. We encode a set of frames such as *Being_employed*, *People_by_origin*, *People_by_religion*, *People_by_age*, *Personal_relationship* that represent the entity relations with respect to human. The testing set is constructed from Wikipedia. Given that Wikipedia provides an abundance pages about people, we extract the sentences that are related to a person's background. We evaluate both the precision and recall with respect to the testing set. Particularly, for precision, it is very likely that multiple frames are extracted for one sentence. We consider the one with the highest score as the best answer. As the next step, we will work on specific domains such as medical text, financial rules, etc. For each domain, it would require the knowledge engineer to create additional frames in order to represent the entity relations that are used there.

5 Current State of Research and Open Issues

Currently, we are working on building the prototype of the system that achieves knowledge annotation and knowledge acquisition. In the first stage, we focus on extracting logical facts from CNL sentences. We have encoded a subset of the frames in FrameNet that suffices to capture the frame relations in one domain. We also run experiments to show the power of KAM in standardizing CNL sentences to logical representations in comparison with other relation extraction tools. As the next step, we will work on extracting rules from CNL text and apply the rules and facts in question answering. Besides, we will expand the LFrame Engine to include additional frames and apply to broader domains.

6 Conclusion

In this paper, we show a novel knowledge acquisition system, KAM. First, it is a new approach in information extraction that can identify English sentences expressing the same meaning in different syntactic forms and standardize them to the same semantic representation. Second, it applies APC, a paraconsistent logical framework to encode English sentences in a logical manner to support inference in the presence of inconsistent knowledge. Third, KAM provides the users an environment to enter and disambiguate the English text and perform question answering tasks.

References

1 Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer, 2007.

2 Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In Alexander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, Third International Conference, CICLing 2002, Mexico City, Mexico, February 17-23, 2002, Proceedings*, volume 2276 of *Lecture Notes in Computer Science*, pages 136–145. Springer, 2002.

3 José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artif. Intell.*, 240:36–64, 2016.

4 Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56, 2014.

5 Broes de Cat, Bart Bogaerts, Maurice Bruynooghe, and Marc Denecker. Predicate logic as a modelling language: The IDP system. *CoRR*, abs/1401.6312, 2014. URL: `http://arxiv.org/abs/1401.6312`.

6 Esra Erdem, Halit Erdogan, and Umut Öztok. BIOQUERY-ASP: querying biomedical ontologies using answer set programming. In Stefano Bragaglia, Carlos Viegas Damásio, Marco Montali, Alun D. Preece, Charles J. Petrie, Mark Proctor, and Umberto Straccia, editors, *Proceedings of the 5th International RuleML2011@BRF Challenge, co-located with the 5th International Rule Symposium, Fort Lauderdale, Florida, USA, November 3-5, 2011*, volume 799 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

7 Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Attempto controlled english for knowledge representation. In Cristina Baroglio, Piero A. Bonatti, Jan Maluszynski, Massimo Marchiori, Axel Polleres, and Sebastian Schaffert, editors, *Reasoning Web, 4th International Summer School 2008, Venice, Italy, September 7-11, 2008, Tutorial Lectures*, volume 5224 of *Lecture Notes in Computer Science*, pages 104–124. Springer, 2008.

8 Tiantian Gao, Paul Fodor, and Michael Kifer. Paraconsistency and word puzzles. *TPLP*, 16(5-6):703–720, 2016.

9 M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub. *Clingo* = ASP + control: Preliminary report. In M. Leuschel and T. Schrijvers, editors, *Technical Communications of the Thirtieth International Conference on Logic Programming (ICLP'14)*, volume arXiv:1405.3694v1, 2014. Theory and Practice of Logic Programming, Online Supplement.

10 Graeme Hirst, David St-Onge, et al. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, 305:305–332, 1998.

11 Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, cmp-lg/9709008, 1997.

12 Richard Johansson and Pierre Nugues. Lth: Semantic structure extraction using nonprojective dependency trees. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 227–230, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

13 Chrstopher R. Johnson, Charles J. Fillmore, Miriam R.L. Petruck, Collin F. Baker, Michael J. Ellsworth, Josef Ruppenhofer, and Esther J. Wood. FrameNet: Theory and Practice, 2002.

14 Paul Kingsbury and Martha Palmer. Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, volume 3. Citeseer, 2003.

15 Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

16 John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi. LCC approaches to knowledge base population at TAC 2010. In *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*. NIST, 2010.

17 Dekang Lin. An information-theoretic definition of similarity. In Jude W. Shavlik, editor, *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998*, pages 296–304. Morgan Kaufmann, 1998.

18 Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 523–534. ACL, 2012.

19 George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

20 Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1341–1351. The Association for Computer Linguistics, 2013.

21 Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

22 Konstantinos Sagonas, Terrance Swift, and David S. Warren. Xsb as an efficient deductive database engine. In *In Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pages 442–453. ACM Press, 1994.

23 Karin Kipper Schuler. *Verbnet: A Broad-coverage, Comprehensive Verb Lexicon*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 2005. AAI3179808.

24 Rolf Schwitter. English as a formal specification language. In *13th International Workshop on Database and Expert Systems Applications (DEXA 2002), 2-6 September 2002, Aix-en-Provence, France*, pages 228–232. IEEE Computer Society, 2002.

25 Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

26 Mihai Surdeanu, David McClosky, Mason R. Smith, Andrey Gusev, and Christopher D. Manning. Customizing an information extraction system to a new domain. In *Proceedings of the ACL 2011 Workshop on Relational Models of Semantics*, RELMS '11, pages 2–10, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

27 Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

28 Zhibiao Wu and Martha Stone Palmer. Verb semantics and lexical selection. In James Pustejovsky, editor, *32nd Annual Meeting of the Association for Computational Linguistics, 27-30 June 1994, New Mexico State University, Las Cruces, New Mexico, USA, Proceedings.*, pages 133–138. Morgan Kaufmann Publishers / ACL, 1994.