Bridging the Gap Between Ontology and Lexicon via Class-Specific Association Rules Mined from a Loosely-Parallel Text-Data Corpus

Basil Ell
CIT-EC, University of Bielefeld, Germany
Department of Informatics, University of Oslo, Norway

Mohammad Fazleh Elahi
CIT-EC, University of Bielefeld, Germany

Philipp Cimiano
CIT-EC, University of Bielefeld, Germany

Abstract

There is a well-known lexical gap between content expressed in the form of natural language (NL) texts and content stored in an RDF knowledge base (KB). For tasks such as Information Extraction (IE), this gap needs to be bridged from NL to KB, so that facts extracted from text can be represented in RDF and can then be added to an RDF KB. For tasks such as Natural Language Generation, this gap needs to be bridged from KB to NL, so that facts stored in an RDF KB can be verbalized and read by humans. In this paper we propose *LexExMachina*, a new methodology that induces correspondences between lexical elements and KB elements by mining class-specific association rules. As an example of such an association rule, consider the rule that predicts that if the text about a person contains the token *"Greek"*, then this person has the relation `nationality` to the entity `Greece`. Another rule predicts that if the text about a `settlement` contains the token *"Greek"*, then this settlement has the relation `country` to the entity `Greece`. Such a rule can help in question answering, as it maps an adjective to the relevant KB terms, and it can help in information extraction from text. We propose and empirically investigate a set of 20 types of class-specific association rules together with different interestingness measures to rank them. We apply our method on a loosely-parallel text-data corpus that consists of data from DBpedia and texts from Wikipedia, and evaluate and provide empirical evidence for the utility of the rules for Question Answering.

2012 ACM Subject Classification Computing methodologies → Information extraction; Computing methodologies → Natural language generation

Keywords and phrases Ontology, Lexicon, Association Rules, Pattern Mining

Digital Object Identifier 10.4230/OASIcs.LDK.2021.33

Supplementary Material *Collection (Dataset and Source Code)*: `http://www.LexExMachina.xyz`

Funding This work has been supported by the EU's Horizon 2020 project Prêt-à-LLOD (grant agreement No 825182) and by the SIRIUS centre: Norwegian Research Council project No 237898.

1 Introduction

There is a fundamental lexical gap between the "names", that is URIs, that are given to data elements in knowledge bases or knowledge graphs on the one hand, and how they are referred to in natural language. Bridging between these two symbol levels is crucial. There are many scenarios in which we need to map from natural language to KB, that is the case for text understanding, information extraction and question answering. There are also scenarios in which we need to map from KB to language, e.g. when verbalizing triples of a knowledge base in natural language [12].

In this paper, we present an approach to inducing correspondences between the lexical and knowledge base level that relies on mining association rules. The association rules that we mine have a lexical or linguistic symbol on the one side, and a KB symbol or structure on the other, thus allowing to bridge between the two levels.

The association rules that we mine are *class-specific* in the sense that at least one of the sides of an association rule expresses a condition that the entities that the association rule talks about belong to a specific class. The motivation for this is that the way a certain property is verbalized depends on the class in question. Similarly, the interpretation of a certain lexical element depends on the context of the class in question. Take the example of the adjective *Greek* that according to classical formal semantics represents a unary predicate, that is a class. When `Greek` modifies a person as in *"Greek politician"*, the correct interpretation with respect to the schema of a knowledge base might be the one that the `nationality` is `Greek`. In case of a city, e.g. *"Greek city"*, the correct interpretation might be that the `country` in which the city is located in is `Greece`. So the interpretation is class-specific. Conversely, take a property such as `author`. In the context of books, the property would be verbalized as *X wrote Y*, while in the context of a music piece the appropriate verbalization would be *X composed Y*.

In this paper we present our approach to mining class-specific association rules from a loosely-parallel dataset consisting of a corpus and corresponding knowledge base. The corpus and KB are loosely parallel in the sense that the text describes the entities in the KB but there is no explicit relation between the two. Further, the relation is not 1:1 in the sense that there are some triples that are not expressed in the text and there are many aspects in the text that are not represented by triples. We describe 20 different types of such class-specific association rules that we mine. We apply our approach to a parallel dataset consisting of the Wikipedia abstracts for 1,297,623 entities from 354 classes, together with the RDF descriptions of these entities. We derive 447,888,109 association rules from this dataset in total. We evaluate our approach on the basis of the well-known QALD (Question Answering over Linked Data) dataset, evaluating in how far our approach can retrieve valid correspondences between lexical and KB elements.

The remainder of this paper is structured as follows: we present our method for mining class-specific association rules in Section 2. We describe the application of our method on a loosely-parallel text-data corpus consisting of texts from Wikipedia and data from DBpedia in Section 3. We present the results of our evaluation on a question answering task in Section 4. Before concluding we discuss related work.

All code and data is available at our website `http://www.LexExMachina.xyz`.

2 Approach

In this section, we describe our approach *LexExMachina*. We introduce relevant preliminaries and notation needed to express the class-specific association rules in Section 2.1. We introduce our approach by an example describing a particular association rule for our motivating example in the introduction in Section 2.2. We describe our general approach in Section 2.3.

2.1 Preliminaries

Let P be a set of (URIs of) properties, let D be a set of documents, let C be a set of classes, let E be a set of entities, let G be an RDF graph, and let L be a set of linguistic patterns (for example, n-grams). Furthermore, let $c_e \subseteq C$ denote classes that entity $e \in E$ belongs to, let $d_e \in D$ denote the document that describes the entity $e \in E$ (e.g., the Wikipedia

article about the entity), and let $l_e \subseteq L$ denote the set of linguistic patterns that occur in the document d_e describing e. An RDF graph is a set of triples of the form (s,p,o) where $s \in \mathcal{U} \cup \mathcal{B}$ is called the triple's subject, $p \in \mathcal{U}$ is called the triple's predicate, and $o \in \mathcal{U} \cup \mathcal{B} \cup \mathcal{L}$ is called the triple's object. $\mathcal{U}$, $\mathcal{B}$, and $\mathcal{L}$ are the sets of URIs, blank nodes, and literals, respectively, and are pairwise disjoint. The set $\mathcal{T}$ of terms is the union of the sets $\mathcal{U}$, $\mathcal{B}$, and $\mathcal{L}$. The sets P, C, and E are true subsets of $\mathcal{U}$.

An association rule has the form $A \Rightarrow B$ where A and B are called events. For example, *Greece occurs in the text* is an event. The support of an event A, denoted by $sup(A)$, is the number of times that this event is true in a given set. For example, given a set of texts, the support of the event *Greece occurs in the text* is the number of documents for which it holds that *Greece occurs in the text*. The confidence of an association rule $A \Rightarrow B$, denoted by $conf(A \Rightarrow B)$, is defined as $conf(A \Rightarrow B) = sup(A \wedge B)/sup(A)$.[1] For example, let B be the event *born occurs in the text*. Thus, the confidence of the rule $A \Rightarrow B$ is the support of the event *Greece and born occur in the text* divided by the support of the event *Greece occurs in the text*. The higher the confidence, the more likely it is that given that a text contains the word *Greece*, it also contains the word *born*. Thus, the confidence of an association rule $A \Rightarrow B$ is identical to the estimated conditional probability $P(B|A)$.

In practice, association rules with high confidence do not necessarily disclose truly interesting event relationships [2]. Therefore, an *interestingness measure* quantifies the interestingness of an association rule. We list the classical null-invariant measures of interestingness as reformulated in terms of estimated conditional probabilities by Wu et al. [16] as well as the null-invariant measure *imbalance ratio* (IR), also introduced by Wu et al. [16]:

$$AllConf(A,B) = min\{P(A|B), P(B|A)\} \tag{1}$$

$$Coherence(A,B) = (P(A|B)^{-1} + P(B|A)^{-1} - 1)^{-1} \tag{2}$$

$$Cosine(A,B) = \sqrt{P(A|B)P(B|A)} \tag{3}$$

$$Kulczynski(A,B) = (P(A|B) + P(B|A))/2 \tag{4}$$

$$MaxConf(A,B) = max\{P(A|B), P(B|A)\} \tag{5}$$

$$IR(A,B) = \frac{|P(A|B) - P(B|A)|}{P(A|B) + P(B|A) - P(A|B) \times P(B|A)} \tag{6}$$

Note that all of these 6 metrics are symmetric, i.e., the order of the events A and B does not matter (e.g., $AllConf(A,B) = AllConf(B,A)$ for any events A and B). The estimated conditional probabilities can be calculated via support counts given the equations $P(B|A) = sup(AB)/sup(A)$ and $P(A|B) = sup(AB)/sup(B)$.

2.2 A Close Look at one Rule Pattern

In this section we describe a rule pattern with the name $c_s, l_s \Rightarrow po$ in detail, before we present the list of all 20 rule patterns in Section 2.3.

Given are a class $c \in C$, a property $p \in P$, a term $o \in \mathcal{T}$, and a linguistic pattern l. Given that an entity e is an instance of the class c and given that the linguistic pattern l occurs in the document d_e that describes the entity e, we want to predict whether the triple (e,p,o) is true. We define two events A and B. AB denotes the conjunction of these two events.

[1] In the remainder of the paper we write AB to denote $A \wedge B$.

$$A = c \in c_e \wedge l \in l_e$$
$$B = c \in c_e \wedge (e, p, o) \in G$$
$$AB = c \in c_e \wedge l \in l_e \wedge (e, p, o) \in G$$

Given a class $c \in C$ and a linguistic pattern l, the support of the event A, denoted by $sup(A)$, can be calculated as $|\{e \in E \mid c \in c_e \wedge l \in l_e\}|$ – thus, the support of the event A is the number of entities where each entity is an instance of the class c and where the linguistic pattern l occurs in the document that describes the entity.

Given a class $c \in C$, a property $p \in P$, and a term $o \in \mathcal{T}$, the support of the event B, denoted by $sup(B)$, can be calculated as $|\{e \in E \mid c \in c_e \wedge (e, p, o) \in G\}|$ – thus, the support of the event B is the number of entities where each entity is an instance of the class c and where the triple (e, p, o) exists in the graph G.

Given a class $c \in C$, a property $p \in P$, a term $o \in \mathcal{T}$, and a linguistic pattern l, the support of the event AB, denoted by $sup(AB)$, can be calculated as $|\{c \in c_e \wedge l \in l_e \wedge (e, p, o) \in G\}|$ – thus, the support of the event AB is the number of entities where each entity is an instance of the class c and where the linguistic pattern l occurs in the document that describes the entity and where the triple (e, p, o) exists in the graph G.

From these events we can construct association rules of the form $A \Rightarrow B$ given a class $c \in C$, a property $p \in P$, a term $o \in \mathcal{T}$, and a linguistic pattern l:

$$c \in c_e \wedge l \in l_e \Rightarrow (e, p, o) \in G$$

For example, with the class $c = dbo{:}Politician$, the property $p = dbo{:}nationality$, the term $o = dbr{:}Greece$, and the linguistic pattern $l = \text{"}Greek\text{"}$, we can create the following association rule:

$$dbo{:}Politician \in c_e \wedge \text{"}Greek\text{"} \in l_e \Rightarrow (e, dbo{:}nationality, dbr{:}Greece) \in G$$

Due to the fact that the linguistic pattern is a 1-gram, matching the pattern against a text is simple enough so that we can calculate the support of the three events via SPARQL queries.[2] Thus, we obtain the values $sup(A) = 128$, $sup(B) = 19$, and $sup(AB) = 19$. The confidence of an association rule of the form $A \Rightarrow B$ can be calculated as $sup(AB)/sup(A) = P(B|A)$. For our example, the confidence of the association rule is $sup(AB)/sup(A) = 19/128 \approx 0.15$.

If the class membership constraints are removed from the event definitions, then we obtain the events $A' = l \in l_e$ and $B' = (e, p, o) \in G$. For the example above, this results in the support values $sup(A') = sup(\text{"}Greek\text{"} \in l_e) = 58,563$, $sup(B') = sup((e, dbo{:}nationality, dbr{:}Greece) \in G) = 464$, and $sup(A'B') = sup(\text{"}Greek\text{"} \in l_e \wedge (e, dbo{:}nationality, dbr{:}Greece) \in G) = 445$, which results in the confidence value of $sup(A'B')/sup(A') = 445/58,563 \approx$

[2] $sup(A)$: `SELECT COUNT(?e) WHERE { ?e rdf:type dbo:Politician . ?e dbo:abstract ?a . FILTER (LANG(?a)="en" && REGEX(?a, "(^|\\ W)Greek(\\ W|$)")) }` → 128; $sup(B)$: `SELECT COUNT(?e) WHERE { ?e rdf:type dbo:Politician . ?e dbo:nationality dbr:Greece }` → 19; $sup(AB)$: `SELECT COUNT(?e) WHERE { ?e rdf:type dbo:Politician . ?e dbo:nationality dbr:Greece . ?e dbo:abstract ?a FILTER(LANG(?a)="en" && REGEX(?a, "(^|\\W)Greek(\\W|$)"))}` → 19. The parts before and after the term `Greek` ensure that the term either occurs at the beginning of the text or after a non-word character and that the term occurs either at the end of the text or is followed by a non-word character. The queries were ran against the public endpoint of DBpedia (`http://dbpedia.org/sparql`) on January 5, 2021.

0.0076, which is significantly lower than the confidence of the association rule *with* class membership constraints (i.e., ≈ 0.15). For this reason, in this paper we only investigate association rules that are class-specific. Note that if the word *Greek* appears in a text about a person, this might indicate that the person is of Greek nationality, whereas if the word *Greek* occurs in a text about a settlement, then this might indicate that the settlement is located in Greece – thus, which property is used depends on the class an entity belongs to.

If for an association rule $A \Rightarrow B$ we have calculated $sup(A)$, $sup(B)$, and $sup(AB)$, then we can not only calculate $P(B|A)$, but also $P(A|B)$, which means that we can calculate the confidence of the "reversed" association rule $B \Rightarrow A$:

$$c \in c_e \wedge (e, p, o) \in G \Rightarrow l \in l_e$$

The name of the reversed rule pattern $c_s, l_s \Rightarrow po$ is $c_s, po \Rightarrow l_s$. For the example above, this is the reversed rule:

$$dbo{:}Politician \in c_e \wedge (e, dbo{:}nationality, dbr{:}Greece) \in G \Rightarrow \text{"}Greek\text{"} \in l_e$$

The confidence of this rule (i.e., P(A|B)) is $sup(AB)/sup(B) = 19/19 = 1$. Given that for an association rule $A \Rightarrow B$ we have computed $P(B|A)$ and $P(A|B)$, we can also compute values for the interestingness measures. Note that because the interestingness measures are symmetric, the interestingness of the rule is the same as the interestingness of the reversed rule for this interestingness measure.

For the example above, with $P(B|A) = 19/128$ and $P(A|B) = 19/19$, we obtain the interestingness measurements $AllConf(A, B) \approx 0.15$, $Coherence(A, B) \approx 0.15$, $Cosine(A, B) \approx 0.39$, $Kulczynski(A, B) \approx 0.57$, $MaxConf(A, B) = 1$, and $IR(A, B) \approx 0.85$.

2.3 Class-specific association rule patterns

The complete set of 20 class-specific association rule patterns is shown in Table 1.

In the rules we have shown above the linguistic pattern occurs anywhere in a text. For the task of deciding whether a text expresses the triple (e_1, r, e_2), one typically regards the string between the mentions of e_1 and e_2 in the text. According to the principle of distant supervision [10], one assumes that a text expresses (e_1, r, e_2) if both entities are mentioned in the text. For example, for the property *dbo:author* the linguistic pattern that appears between the mentions of the arguments could be *is the author of* or *is best known for her*. Thus, we present rule patterns where the linguistic patterns that are made use of do not occur anywhere in a text but instead need to occur between the arguments of a relation. We refer to these rule patterns as *localized rule patterns* and to the rules where linguistic patterns can occur anywhere in the text as *non-localized* rule patterns. Note that because localization is predicate-specific, rule patterns that do not specify a predicate cannot be localized.

Let $l_e^{c,p,d}$ denote the set of linguistic patterns that occur in the document d_e that describes the entity e where e is an instance of the class c and where the linguistic patterns occur between the arguments of the relation p. The arguments of the relation appear in the order d, which is either *so* (*subject then object*), or *os* (*object then subject*).

The following localized rule predicts a property-object pair for an entity where in the text about the entity a linguistic pattern occurs that has been found between arguments of this relation in other text about entities of the same class:

$$dbo{:}Settlement \in c_e \wedge \text{"the Metropolitan City of Turin"} \in l_e^{dbo:Settlement,dbo:region,so}$$
$$\Rightarrow (e, dbo{:}region, dbr{:}Piedmont) \in G$$

Table 1 The list of 20 class-specific association rule patterns.

$c \in c_e \wedge l \in l_e \Rightarrow (e,p,o) \in G$	$(c_s, l_s \Rightarrow po)$
$c \in c_e \wedge l \in l_e^{c,p,d} \Rightarrow (e,p,o) \in G$	$(c_s, ll_s \Rightarrow po)$
$c \in c_e \wedge l \in l_e \Rightarrow \exists o \in \mathcal{T} : (e,p,o) \in G$	$(c_s, l_s \Rightarrow p)$
$c \in c_e \wedge l \in l_e^{c,p,d} \Rightarrow \exists o \in \mathcal{T} : (e,p,o) \in G$	$(c_s, ll_s \Rightarrow p)$
$c \in c_e \wedge l \in l_e \Rightarrow \exists p \in \mathcal{U} : (e,p,o) \in G$	$(c_s, l_s \Rightarrow o)$
$c \in c_e \wedge l \in l_e \Rightarrow (s,p,e) \in G$	$(c_o, l_o \Rightarrow sp)$
$c \in c_e \wedge l \in l_e^{c,p,d} \Rightarrow (s,p,e) \in G$	$(c_o, ll_o \Rightarrow sp)$
$c \in c_e \wedge l \in l_e \Rightarrow \exists p \in \mathcal{U} : (s,p,e) \in G$	$(c_o, l_o \Rightarrow s)$
$c \in c_e \wedge l \in l_e \Rightarrow \exists s \in \mathcal{U} \cup \mathcal{B} : (s,p,e) \in G$	$(c_o, l_o \Rightarrow p)$
$c \in c_e \wedge l \in l_e^{c,p,d} \Rightarrow \exists s \in \mathcal{U} \cup \mathcal{B} : (s,p,e) \in G$	$(c_o, ll_o \Rightarrow p)$
$c \in c_e \wedge (e,p,o) \in G \Rightarrow l \in l_e$	$(c_s, po \Rightarrow l_s)$
$c \in c_e \wedge (e,p,o) \in G \Rightarrow l \in l_e^{c,p,d}$	$(c_s, po \Rightarrow ll_s)$
$c \in c_e \wedge \exists o \in \mathcal{T} : (e,p,o) \in G \Rightarrow l \in l_e$	$(c_s, p \Rightarrow l_s)$
$c \in c_e \wedge \exists o \in \mathcal{T} : (e,p,o) \in G \Rightarrow l \in l_e^{c,p,d}$	$(c_s, p \Rightarrow ll_s)$
$c \in c_e \wedge \exists p \in \mathcal{U} : (e,p,o) \in G \Rightarrow l \in l_e$	$(c_s, o \Rightarrow l_s)$
$c \in c_e \wedge (s,p,e) \in G \Rightarrow l \in l_e$	$(c_o, sp \Rightarrow l_o)$
$c \in c_e \wedge (s,p,e) \in G \Rightarrow l \in l_e^{c,p,d}$	$(c_o, sp \Rightarrow ll_o)$
$c \in c_e \wedge \exists p \in \mathcal{U} : (s,p,e) \in G \Rightarrow l \in l_e$	$(c_o, s \Rightarrow l_o)$
$c \in c_e \wedge \exists s \in \mathcal{U} \cup \mathcal{B} : (s,p,e) \in G \Rightarrow l \in l_e$	$(c_o, p \Rightarrow l_o)$
$c \in c_e \wedge \exists s \in \mathcal{U} \cup \mathcal{B} : (s,p,e) \in G \Rightarrow l \in l_e^{c,p,d}$	$(c_o, p \Rightarrow ll_o)$

In this example, $l_e^{dbo:Settlement,dbo:region,so}$ is the set of linguistic patterns that occur in d_e and that frequently occur in texts about instances of the class *dbo:Settlement* between the arguments of the relation *dbo:region* where these arguments appear in the order *subject then object*.

3 Mining class-specific association rules from Wikipedia and DBpedia as loosely-parallel corpus

The loosely-parallel text-data corpus we use consists of seven files[3] from the English DBpedia [1]. We refer to it as a loosely-coupled text-data corpus because this data contains the short abstracts of Wikipedia articles as well as structured data extracted from DBpedia. The information that is contained in the DBpedia files has not been extracted from the article's natural language text, which means that not every piece of information contained in

[3] From `https://wiki.dbpedia.org/develop/datasets` we retrieved the following files in the stated versions: infobox-properties_lang=en.ttl.bz2 (v2020.11.01), instance-types_lang=en_specific.ttl.bz2 (v2020.12.01), mappingbased-literals_lang=en.ttl.bz2 (v2020.12.01), mappingbased-objects_lang=en.ttl.bz2 (v2020.12.01), short-abstracts_lang=en.ttl.bz2 (v2020.07.01), labels_lang=en.ttl.bz2 (v2020.12.01), and anchor-text_lang=en.ttl.bz2 (v2020.12.01). Labels and anchors were only used to identify the arguments of a relation so that localized linguistic patterns can be collected. That means that *rdf:type* and *rdfs:label* never occur as predicate in any rule that we have mined.

an article is contained in a DBpedia file. Furthermore, not every piece of information that is contained in a DBpedia file is expressed in a Wikipedia article's short abstract (e.g., an athlete's height is usually only contained in a table and is not expressed in the text).

By restricting a class to have at least 100 instances and ignoring *owl:Thing*, we obtained a set of 354 classes ($min\_entities\_per\_class = 100$). For each class, we randomly selected at most 10,000 instances ($max\_entities\_per\_class = 10,000$). In total, we selected 1,297,623 entities, which amounts to approximately 22.63% of all entities for which an abstract exists.

We tokenized the abstract of each entity by splitting at whitespaces and then removed the characters dot ('.'), comma (','), round brackets ('(' and ')'), and colon (':'). From the obtained token sequences we extracted those n-grams ($n \in [1..5]$) that contain at least one non-stopword – we used the NLTK stopword list,[4] which contains 127 entries. We discarded those 1-grams that consist of less than four characters ($min\_onegram\_length = 4$).

For the localized property patterns, we carried out a simple form of coreference resolution, replacing the pronouns *he*, *she*, and *it* with the entity's *rdfs:label*.

For patterns to be localized, the arguments of a relation need to be detected. For this purpose we make use of an entity's *rdfs:label* as well as those anchor texts that refer at least 10 times to a given entity ($min\_anchor\_count = 10$). We also try to identify literal values. We convert literals of type *xsd:date* into a natural language representation such as *2021-03-21*^^*xsd:date* to *21 March 2021*, but leave literals with other datatypes unchanged. If both arguments of a relation were detected and the length of the string between the arguments is not higher than 100 characters ($max\_propertystring\_length = 100$) and consists of at least 5 characters ($min\_propertystring\_length = 5$), we tokenized the string and extract n-grams ($n \in [1, 5]$) as described above. For each pattern, we recorded in which order the arguments occurred in the text (i.e., $d \in \{so, os\}$).

The set of linguistic patterns for a class is the set of all n-grams that were found for at least 5 instances of the class ($min\_pattern\_count = 5$). For the localized property patterns, a pattern had to occur for at least 5 instances of the class ($min\_propertypattern\_count = 5$) for each combination of class and property and order of arguments. This means that the rules have, depending on which side the linguistic pattern occurs, a value for $sup(A)$ or $sup(B)$ of greater or equal to 5.

Given the parameter settings above, we obtained 447,888,109 rules – 427,541,617 non-localized rules and 20,346,492 localized rules. The number of rules found for each rule pattern is shown in Table 2. Note that we set rather low threshold values as this allows to extract data for higher threshold values by filtering, instead of mining, and to find appropriate threshold parameters (e.g., for $sup(A)$, $sup(B)$, $sup(AB)$, $P(B|A)$, $P(A|B)$). For a particular linguistic pattern, i.e., the token "Greek", Table 3 shows the 20 localized rules that are ranked highest according to the Cosine interestingness measure. These rules contain the linguistic pattern on any side of the association rule.

4 Evaluation

We evaluate the utility of the rules that we have mined in the context of the task of Question Answering over an RDF knowledge base. Given a natural language question and an RDF knowledge base, typically, the goal is to infer a SPARQL query that represents the meaning of the question using the KB's vocabulary, so that evaluating the query on the KB results in the KB's answer(s) to the question. We created a corpus of (question, query) pairs from the

[4] The list of stopwords is available at `https://gist.github.com/sebleier/554280` (Accessed 2021-02-20).

Table 2 The number of rules found for each rule pattern.

Group of rule patterns		Number of rules
$c_s, po \Rightarrow l_s$	$c_s, l_s \Rightarrow po$	75,127,937 each
$c_s, po \Rightarrow ll_s$	$c_s, ll_s \Rightarrow po$	4,500,459 each
$c_s, p \Rightarrow l_s$	$c_s, l_s \Rightarrow p$	98,317,655 each
$c_s, p \Rightarrow ll_s$	$c_s, ll_s \Rightarrow p$	5,293,226 each
$c_s, o \Rightarrow l_s$	$c_s, l_s \Rightarrow o$	67,147,957 each
$c_o, sp \Rightarrow l_o$	$c_o, l_o \Rightarrow sp$	3,812,313 each
$c_o, sp \Rightarrow ll_o$	$c_o, ll_o \Rightarrow sp$	157,519 each
$c_o, s \Rightarrow l_o$	$c_o, l_o \Rightarrow s$	429,627 each
$c_o, p \Rightarrow l_o$	$c_o, l_o \Rightarrow p$	6,499,288 each
$c_o, p \Rightarrow ll_o$	$c_o, ll_o \Rightarrow p$	222,042 each
		447,888,109 total

Table 3 The top-20 localized rules that contain the linguistic pattern *Greek*, ordered by the Cosine interestingness measure. We abbreviated *dbo*:$FormerMunicipality$ to *dbo*:FM.

Cos	Rule
0.9	$dbo{:}Model \in c_e \wedge (e, dbp{:}birthPlace, dbr{:}Greece) \in G \Rightarrow$ "Greek" $\in l_e^{dbo:Model,p,so}$
0.9	$dbo{:}Model \in c_e \wedge$ "Greek" $\in l_e^{dbo:Model,p,so} \Rightarrow (e, dbp{:}birthPlace, dbr{:}Greece) \in G$
0.88	$dbo{:}RugbyClub \in c_e \wedge (e, dbo{:}location, dbr{:}Greece) \in G \Rightarrow$ "Greek" $\in l_e^{dbo:RugbyClub,dbo:location,so}$
0.88	$dbo{:}RugbyClub \in c_e \wedge$ "Greek" $\in l_e^{dbo:RugbyClub,dbo:location,so} \Rightarrow (e, dbo{:}location, dbr{:}Greece) \in G$
0.88	$dbo{:}Model \in c_e \wedge (e, dbo{:}birthPlace, dbr{:}Greece) \in G \Rightarrow$ "Greek" $\in l_e^{dbo:Model,dbo:birthPlace,so}$
0.88	$dbo{:}Model \in c_e \wedge$ "Greek" $\in l_e^{dbo:Model,dbo:birthPlace,so} \Rightarrow (e, dbo{:}birthPlace, dbr{:}Greece) \in G$
0.87	$dbo{:}FormerMunicipality \in c_e \wedge (e, dbo{:}country, dbr{:}Greece) \in G \Rightarrow$ "Greek" $\in l_e^{dbo:FM,dbo:country,so}$
0.87	$dbo{:}FM \in c_e \wedge (e, dbo{:}type, dbr{:}Prefectures\_of\_Greece) \in G \Rightarrow$ "Greek" $\in l_e^{dbo:FM,dbo:country,so}$
0.87	$dbo{:}FM \in c_e \wedge (e, dbp{:}subdivisionName, dbr{:}Greece) \in G \Rightarrow$ "Greek" $\in l_e^{dbo:FM,dbp:subdivisionName,so}$
0.87	$dbo{:}FM \in c_e \wedge$ "Greek" $\in l_e^{dbo:FM,dbo:country,so} \Rightarrow (e, dbo{:}country, dbr{:}Greece) \in G$
0.87	$dbo{:}FM \in c_e \wedge$ "Greek" $\in l_e^{dbo:FM,dbo:type,so} \Rightarrow (e, dbo{:}type, dbr{:}Prefectures\_of\_Greece) \in G$
0.87	$dbo{:}FM \in c_e \wedge$ "Greek" $\in l_e(c, p, so) \Rightarrow (e, dbp{:}subdivisionName, dbr{:}Greece) \in G$
0.83	$dbo{:}President \in c_e \wedge (e, dbo{:}nationality, dbr{:}Greece) \in G \Rightarrow$ "Greek" $\in l_e^{dbo:President,dbo:nationality,so}$
0.83	$dbo{:}President \in c_e \wedge$ "Greek" $\in l_e^{dbo:President,dbo:nationality,so} \Rightarrow (e, dbo{:}nationality, dbr{:}Greece) \in G$
0.82	$dbo{:}Swimmer \in c_e \wedge (e, dbo{:}birthPlace, dbr{:}Greece) \in G \Rightarrow$ "Greek" $\in l_e^{dbo:Swimmer,dbo:birthPlace,so}$
0.82	$dbo{:}Swimmer \in c_e \wedge$ "Greek" $\in l_e^{dbo:Swimmer,dbo:birthPlace,so} \Rightarrow (e, dbo{:}birthPlace, dbr{:}Greece) \in G$
0.82	$dbo{:}Model \in c_e \wedge (e, dbp{:}birthPlace, dbr{:}Greece) \in G \Rightarrow$ "Greek" $\in l_e^{dbo:Model,dbp:birthPlace,os}$
0.82	$dbo{:}RugbyClub \in c_e \wedge (e, dbp{:}location, dbr{:}Greece) \in G \Rightarrow$ "Greek" $\in l_e^{dbo:RugbyClub,dbp:location,so}$
0.82	$dbo{:}Model \in c_e \wedge$ "Greek" $\in l_e^{dbo:Model,dbp:birthPlace,os} \Rightarrow (e, dbp{:}birthPlace, dbr{:}Greece) \in G$
0.82	$dbo{:}RugbyClub \in c_e \wedge$ "Greek" $\in l_e^{dbo:RugbyClub,dbp:location,so} \Rightarrow (e, dbp{:}location, dbr{:}Greece) \in G$

QALD (Question Answering over Linked Data)[5] challenge series[6] that consists of 601 pairs. For each (question, query) pair (t,q), we tokenize t and create a set of linguistic patterns in the same way as we have processed the abstracts and extracted patterns, explained in Section 3. For each query q we create the (possibly empty) sets s_q, p_q, o_q, sp_q, and po_q that are defined as follows. s_q is the set of terms that occur in subject position of triple patterns in q, p_q is the set of terms that occur in predicate position of triple patterns in q, o_q is the set of terms that occur in object position of triple patterns in q, sp_q is a set of tuples of the form (t_1, t_2) where q contains a triple pattern with t_1 in subject position and t_2 in predicate position, and po_q is a set of tuples of the form (t_1, t_2) where q contains a triple pattern with t_1 in predicate position and t_2 in object position. From the set p_q we removed the term *rdfs:label* and the term *rdf:type*, and from the sets sp_q and po_q we removed all pairs of terms that contained the term *rdfs:label* or the term *rdf:type*, because in the experiment we decided against learning rules that are class-specific and that mention another type or that predict a label, although this might be included in the future. q_s was non-empty for 315 queries, q_p was non-empty for 579 queries, q_o was non-empty for 322 queries, q_{sp} was non-empty for 311 queries, and q_{po} was non-empty for 229 queries. 275 distinct terms occurred in subject position, 298 distinct terms occurred in predicate position, 296 distinct terms occurred in object position, 309 distinct term pairs occurred in subject-predicate position, and 259 distinct term pairs occurred in predicate-object position.

As an example, consider the following SPARQL query which corresponds to the question *Give me English actors starring in Lovesick.*[7]

```
SELECT DISTINCT ?uri WHERE {
    res:Lovesick dbo:starring ?uri .
    { ?uri dbo:birthPlace res:England . }
    UNION
    { ?uri rdf:type yago:EnglishFilmActors . }
}
```

Given the SPARQL query above the sets have the following content: $s_q = \{res{:}Lovesick\}$, $p_q = \{dbo{:}starring, dbo{:}birthPlace\}$, $o_q = \{res{:}England, yago{:}EnglishFilmActors\}$, $sp_q = \{(res{:}Lovesick, dbo{:}starring)\}$, $po_q = \{(dbo{:}birthPlace, res{:}England)\}$. The set of linguistic patterns l_q contains the 1-grams *"actors"*, *"Give"*, *"English"*, *"Lovesick"*, and *"starring"*, the 2-grams *"Give me"*, *"actors starring"*, *"me English"*, *"in Lovesick"*, *"starring in"*, and *"English actors"*, and so forth up to 5-grams.

Given a (question, query) pair, we can now find all rules for the 10 rule patterns $c_s, l_s \Rightarrow po$; $c_s, ll_s \Rightarrow po$; $c_s, l_s \Rightarrow p$; $c_s, ll_s \Rightarrow p$; $c_s, l_s \Rightarrow o$; $c_o, l_o \Rightarrow sp$; $c_o, ll_o \Rightarrow sp$; $c_o, l_o \Rightarrow s$; $c_o, l_o \Rightarrow p$; and $c_o, ll_o \Rightarrow p$, i.e., those that predict KB terms based on linguistic patterns. For all these rule patterns, a triple pattern occurs on the right side of the association rules. For a rule r, s_r denotes the triple pattern's subject term, p_r denotes the triple pattern's predicate term, and o_r denotes the triple pattern's object term.

[5] See `http://qald.aksw.org/`

[6] We used all files containing (question, query) pairs from the QALD challenge series that we could get hold on. We used the files `dbpedia-test.xml` and `dbpedia-train.xml` from QALD-1, QALD-2, and QALD-3, the files `qald-4_multilingual_test.xml` and `qald-4_multilingual_train.xml` from QALD-4, the file `qald-5_train.xml` from QALD-5, the files `qald-6-test-multilingual.json` and `qald-6-train-multilingual.json` from QALD-6, the file `qald-7-train-multilingual.json` from QALD-7, and the file `qald-9-train-multilingual.json`. In the case where a question appeared in several challenges we only make use of the corresponding query from the most recent challenge.

[7] The example is taken from the QALD-5 challenge, question #293, file *qald-5_train.xml*.

Let R be a set of rules and let Q be a set of (question, query) pairs. Given a set of rules R and a query q, the set of true positives for predicate terms, denoted by $TP_p(q, R)$, is the set of terms that are necessary for building the query (i.e., those terms that exist in predicate position in the query) and that are proposed by some rule $r \in R$. Likewise, we can define TP_s, TP_o, TP_{sp}, and TP_{po}. The set $FP_p(q, R)$ of false positives for predicate terms is the set of terms that are incorrectly proposed as necessary for building the query (i.e., those terms that exist in predicate position in the query) and that are proposed by some rule $r \in R$. Likewise, we can define FP_s, FP_o, FP_{sp}, and FP_{po}. The set $FN_p(q, R)$ of false negatives for predicate terms is the set of terms that are necessary for building the query (i.e., those terms that exist in predicate position in the query) but are not proposed by any rule $r \in R$. Likewise, we can define FN_s, FN_o, FN_{sp}, and FN_{po}. Given TP_x, FP_x, and FN_x, we can calculate micro-averaged precision ($micro{-}P_x(Q, R)$), micro-averaged recall ($micro{-}R_x(Q, R)$), micro-averaged $F1$ ($micro{-}F1_x(Q, R)$), macro-averaged precision ($macro{-}P_x(Q, R)$), macro-averaged recall ($macro{-}R_x(Q, R)$), and macro-averaged $F1$ ($macro{-}F1_x(Q, R)$) for each prediction type $x \in \{s, p, o, sp, po\}$.

Within the set of non-localized rules we found 17,165,819[8] rules that contain a linguistic pattern that appears in a QALD question. In the set of localized rules we found 742,891[9] rules that contain a linguistic pattern that appears in a QALD question. From these rules, only for 128,223 (≈1%) non-localized rules and for 42,838 (≈6%) localized rules there exists a (question, query) pair such that the rule contains a linguistic pattern that exists in the question and the rule predicts a term or a pair of terms that occurs in the query – thus, these are the desired/helpful rules.[10]

Without filtering the set R of rules, we measured the recall values, because these help us to understand the upper bounds for recall for any subset of R. For the set of non-localized (localized) rules, we measured the following values: $micro{-}R_s$=0.08 (0.03), $microR_p$=0.92 (0.74), $micro{-}R_o$=0.31 (0.21), $micro{-}R_{sp}$=0.02 (0.01), $micro{-}R_{po}$=0.47 (0.3), $macro{-}R_s$=0.08 (0.02), $macro{-}R_p$=0.92 (0.71), $macroR_o$=0.27 (0.16), $macro{-}R_{sp}$=0.02 (0), and $macro{-}R_{po}$ =0.44 (0.26). All precision values were close to zero. It can be seen that the localized rules do not perform better than the non-localized rules.

We investigated the impact of the individual parameters on precision, recall and $F1$ for non-localized and for localized rules. We filtered R with $sup(A), sup(B) \in \{5, 10, 15, 20\}$, $sup(AB) \in \{5, 10, 15\}$, $P(B|A), P(A|B) \in \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$, and for the *AllConf* measure the threshold values $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. Instead of exploring the cartesian product of possible parameter value combinations, for each experiment we only let one parameter take a value that is not the lowest possible value, which results in a set of 28 experiments. For localized rules, Figure 1 shows the precision values for each experiment, Figure 2 shows the recall values for each experiment, and Figure 3 shows the $F1$ values

[8] $c_s, l_s \Rightarrow o$: 5,599,910, $c_o, l_o \Rightarrow p$: 529,331, $c_s, l_s \Rightarrow p$: 3,828,243, $c_s, l_s \Rightarrow po$: 6,395,776, $c_o, l_o \Rightarrow s$: 59,584, $c_o, l_o \Rightarrow sp$: 752,974

[9] $c_o, ll_o \Rightarrow sp$: 41,204, $c_o, ll_o \Rightarrow p$: 30,870, $c_s, ll_s \Rightarrow po$: 487,176, $c_s, ll_s \Rightarrow p$: 409,408

[10] Objects were correctly predicted by 8,044 rules of type $c_s, l_s \Rightarrow o$, 16,207 rules of type $c_s, l_s \Rightarrow po$, and 6,625 rules of type $c_s, ll_s \Rightarrow po$; predicates were correctly predicted by 5,005 rules of type $c_o, l_o \Rightarrow p$, 25,127 rules of type $c_s, l_s \Rightarrow p$, 107,186 rules of type $c_s, l_s \Rightarrow po$, 15,626 rules of type $c_o, l_o \Rightarrow sp$, 1,384 rules of type $c_o, ll_o \Rightarrow p$, 10,392 rules of type $c_s, ll_s \Rightarrow p$, 24,709 rules of type $c_s, ll_s \Rightarrow po$, and 1,571 rules of type $c_o, ll_o \Rightarrow sp$; subjects were correctly predicted by 16 rules of type $c_o, l_o \Rightarrow s$, 100 rules of type $c_o, l_o \Rightarrow sp$, and 37 rules of type $c_o, ll_o \Rightarrow sp$; subject-predicate pairs were correctly predicted by 9 rules of type $c_o, l_o \Rightarrow sp$ and 2 rules of type $c_o, ll_o \Rightarrow sp$; property-object pairs were correctly predicted by 3,173 rules of type $c_s, l_s \Rightarrow po$ and by 1,577 rules of type $c_s, ll_s \Rightarrow po$.

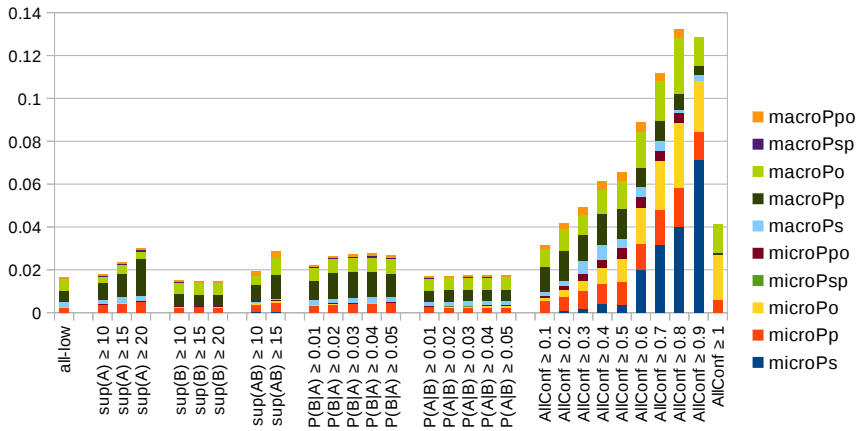

Figure 1 Precision values for each of the 28 experiments with localized rules.

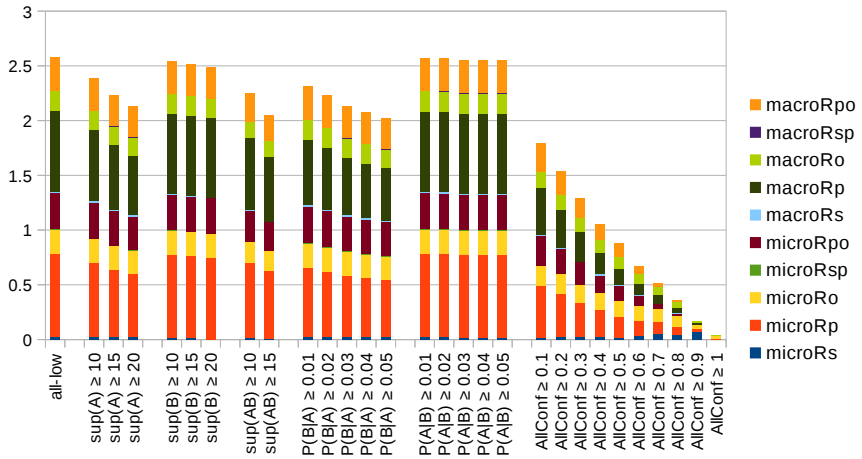

Figure 2 Recall values for each of the 28 experiments with localized rules.

for each experiment. The interestingness threshold appears to have the highest impact on precision, recall, and $F1$. However, increasing the threshold for the *AllConf* measure also decreases precision. Note that due to the bar chart being stacked, recall values can be above 1, because each recall value, e.g., $macro{-}R_{po}$, is a value in the range $[0, 1]$.

4.1 Gold Standard Evaluation

The evaluation described previously considers all possible pairs of lexical elements and KB elements that can be extracted from pairs of NL question and SPARQL queries in the QALD dataset. In order to allow for a more controlled evaluation that allows us to examine the performance of our approach on different parts-of-speech, we manually created a gold standard from QALD-9 for three parts-of-speech: for adjectives referring to a pair of property and object, for verbs referring to a property, and for (relational) nouns referring to a property. We describe the gold standards for the three different parts-of-speech in the following:

- Gold standard for adjectives: comprising of 13 adjectives referring to a pair of property and object. As an example, the adjective *Swedish* in the question *"Give me all Swedish holidays"* refers to the pair $(dbo{:}country, res{:}Sweden)$.

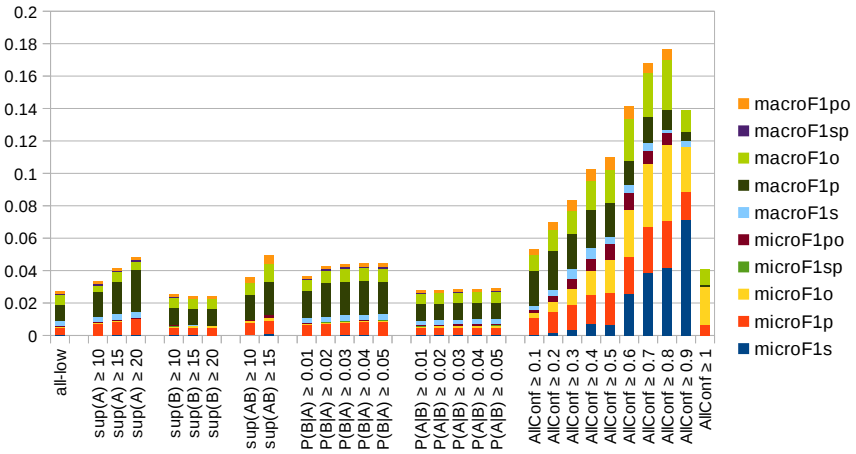

Figure 3 $F1$ values for each of the 28 experiments with localized rules.

- Gold standard for verbs: comprising of 69 verbs referring to a property. As an example, the verb *dissolve* in the question *"When did the Ming dynasty dissolve?"* refers to the property *dbo*:*dissolutionDate*.
- Gold standard for (relational) nouns: comprising of 55 nouns. As an example, the relational noun *founder (of)* refers to the property *dbo*:$founder$ in the question *"Who is the founder of Penguin Books?"*.

In Table 4, we give the results in terms of four metrics: MRR, Hits@1, Hits@5, Hits@10. Mean reciprocal rank (MRR) is a measure used in information retrieval to evaluate ranked lists of results. The MRR is defined as follows:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{|rank_i|}$$

In our case the query is the lexical element in question and the retrieved list corresponds to the KB elements ranked by the corresponding interestingness measure. Hits@k denotes the percentage of queries for which the correct KB element is within the top k results. We provide the results for the best configuration in terms of hyperparameters for each part-of-speech.

The best results were obtained for adjectives when we filtered rules that do not satisfy the following constraints: $supA \geq 5$, $supB \geq 50$, $P(A|B) \geq 0.1$, $P(B|A) \geq 0.05$, and an interestingness value ≥ 0.2 Among the interestingness measures, $MaxConf$ achieves higher performance (0.23, 0.35, 0.35, and 0.4 for MRR, Hits@1 and Hits@5, Hits@10 respectively) than all other interestingness measures. The low results in terms of MRR are due to the fact that in some cases, the correct (property, object) pair for an adjective is ranked rather low in the list. For the adjective *Canadian*, the correct pair (*dbo*:*country*, *dbr*:*Canada*) ranks at position 17 of the best ranking with the $MaxConf$ measure, while other related (property, object) pairs that are more specific rank higher, such as (*dbo*:*region*, *dbr*:*Saskatchewan*), (*dbo*:*location*, *dbr*:*Ontario*) etc. The best results were obtained for verbs with the configuration $supA \geq 50$, $supB \geq 50$, $P(A|B) \geq 0.1$, $P(B|A) \geq 0.1$, MaxConf ≥ 0.2. For the majority of verbs including *create*, *design*, *develop*, *die*, *direct*, *found*, *marry*, etc. the corresponding correct property *dbo*:*creator*, *dbo*:*designer*, *dbo*:*developer*, *dbo*:*deathPlace*, *dbo*:*director*, *dbo*:$founder$, *dbo*:*spouse* rank at position 1. The best results were obtained for relative nouns with the configuration $supA \geq 50$, $supB \geq 50$, $P(A|B) \geq 0.1$, $P(B|A) \geq 0.05$, IR ≥ 0.2.

Table 4 Results of Gold Standard Evaluation of three parts-of-speech: adjective, verb, and noun.

Measure	Adjective				Verb				Noun			
	MRR	Hits1	Hits5	Hits10	MRR	Hits1	Hits5	Hits10	MRR	Hits1	Hits5	Hits10
Cosine	0.08	0.05	0.2	0.2	0.28	0.25	0.31	0.31	0.19	0.18	0.2	0.2
Coherence	0.05	0.0	0.15	0.2	0.28	0.25	0.31	0.31	0.19	0.18	0.2	0.2
AllConf	0.04	0.0	0.15	0.2	0.28	0.25	0.31	0.31	0.19	0.18	0.2	0.2
MaxConf	**0.23**	**0.35**	**0.35**	**0.4**	**0.31**	**0.31**	**0.31**	**0.31**	0.19	0.18	0.2	0.2
IR	0.13	0.05	0.2	0.4	0.31	0.31	0.31	0.31	**0.2**	**0.2**	**0.2**	**0.2**
Kulczynski	0.11	0.05	0.2	0.3	0.28	0.25	0.31	0.31	0.19	0.18	0.2	0.2

5 Related Work

Related work can be grouped into two areas: i) (mining of patterns for) information extraction from text to RDF, and ii) (mining of patterns for) natural language generation from RDF.

Several works, such as by Gerber et al. [7], Nakashole et al. [13], and Walter et al. [15], apply the distant supervision principle to extract relation-specific patterns from natural language sentences. In our framework, relation-specific patterns can be expressed with the association rule patterns $c_s, ll_s \Rightarrow p$ and $c_o, ll_o \Rightarrow p$.

Gerber et al. [7] apply their approach to texts from Wikipedia and DBpedia as KB. The patterns, called BOA patterns, can be used to extract relations from texts and to populate a knowledge base with the extraction results. BOA patterns are scored based on support, as we propose as well, but furthermore BOA patterns are scored on typicity and specificity, whereas we make use of conditional probabilities and interestingness measures. An example of a BOA pattern for the predicate *subsidiary* is *?D?'s acquisition of ?R?*. Here, *?D?* and *?R?* matches entities that are instances of the classes specified as the domain and range of the predicate, respectively. Thus, a BOA pattern can be specific to up to two classes.

Nakashole et al. [13] introduce SOL patterns. These patterns can consist of syntactic features, ontological type signatures, and lexical features. In contrast to our approach, the authors extract patterns from dependency-parsed sentences instead of from tokenized texts and collect dependency paths between identified entities. Patterns are scored by support and confidence. An example of an SOL pattern for the relation *hasMusicalIdol* is *<musician> PRP idol <musician>*, where *musician*, the ontological type signature, matches any entity that is an instance of the class *musician* and *PRP* matches any token that is a pronoun.

The approach M-ATOLL by Walter et al. [15] mines textual patterns that denote binary relations between entities. The text corpus is dependency-parsed and natural language patterns are identified via a set of manually defined dependency graph patterns that are matched against the parsed text. The resulting patterns are represented in *lemon* [9] format. Going beyond M-ATOLL, we do not rely on a pre-definend set of patterns, but mine the patterns inductively from data (that has not been dependency-parsed).

In contrast to the previous three approaches, although also extracting relations from Wikipedia abstracts and making use of the distant supervision principle, Heist et al. [8] propose an approach that does not make use of linguistic features, for example by considering the position of an identified entity in an abstract. The authors train several classification algorithms and show that a classifier trained on one language can also classify relations in another language, which is possible since the features aren't language-specific in the sense that they do not make use of lexical or syntactic information.

Ding et al. [4] propose an approach to map adjectives to existential restrictions over a KB. Their approach, Adj2ER, finds for example that the adjective *American* can be expressed via the existential restriction $\exists dbo{:}nationality.\{dbr{:}United\_States\}$. This existential restriction

is comparable to the rule pattern $c_s, l_s \Rightarrow po$. As a further similarity, the authors take into account which class an adjective modifies. Adj2ER can create existential restrictions that contain negations. For example, the approach finds that for instances of the class *Actor* the adjective *alive* can be mapped to $\neg\exists deathDate.\top$. Negation cannot be expressed within our framework of association rules. Instead of distant supervision on natural language text, for an adjective and a class their approach collects entities that are instances of that class and then create two sets: one set where the instance and the adjective co-occur in some text and the other set of entities that do not. Then, they make use of the information in a KB about these entities to derive the existential restrictions.

A simple form of generation of natural language text from RDF can be realized, as Sun and Mellish [14] show, by categorizing the names of terms such as predicates (e.g., "has" + noun) and by making use of a few templates specific to these categories. The approach requires the names in an ontology to follow certain conventions and creates verbalizations that may not always be natural. Moreover, each triple is verbalized as an individual sentence. A possibility to create verbalizations that are natural in style is to make use of a lexicon, as shown by Cimiano et al. [3]. However, such a lexicon may not always be available. Ell and Harth [5] present an approach that applies the distant supervision principle and automatically extracts verbalization templates that express multiple triples in one sentence. A good overview about NLG from RDF can be found in the context of the WebNLG challenge[11] [6]. Approaches that tackle this challenge need to be able to carry out tasks such as sentence segmentation, lexicalization, aggregation, and surface realisation. Those association rules mined by our approach that predict a linguistic pattern could be applicable in the context of the lexicalization task. Recent work by Moussallem et al. [11] presents an approach based on a encoder-decoder architecture that is capable of generating multilingual verbalizations.

6 Conclusion

We have presented *LexExMachina*, a new approach to closing the gap between lexicon and ontology by mining a set of 20 types of class-specific association rules that connect a lexical element to a data element from a KB. These rules can be used for information extraction, question answering as well as KB verbalization tasks. We have mined association rules from the loosely-parallel corpus consisting of Wikipedia and DBpedia for the 354 classes that have at least 100 instances. The resulting rules have been evaluated on a QA task of reconstructing all the elements of the query from the NL question by relying on these correspondences as well as on a manually created gold standard that allows us to inspect the results for different parts-of-speech. Our framework subsumes many of the pattern mining approaches proposed so far and shows promising results. Although our experiment showed that high-quality and high-coverage association rules can be found, for example those that contain the token *Greek*, shown in Table 3, we need to investigate further how to increase precision without severely sacrificing recall. Beyond the seven parameters taken into account so far, we plan to investigate the impact of further parameters, such as the length of a string between two arguments from which the patterns are extracted, and how fuzzy matching can help to increase recall.

[11] See `https://webnlg-challenge.loria.fr/`

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 Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond Market Baskets: Generalizing Association Rules to Correlations. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 265–276, 1997.

3 Philipp Cimiano, Janna Lüker, David Nagel, and Christina Unger. Exploiting ontology lexica for generating natural language texts from RDF data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 10–19, 2013.

4 Jiwei Ding, Wei Hu, Qixin Xu, and Yuzhong Qu. Mapping Factoid Adjective Constraints to Existential Restrictions over Knowledge Bases. In *ISWC*, pages 164–181. Springer, 2019.

5 Basil Ell and Andreas Harth. A language-independent method for the extraction of rdf verbalization templates. In *Proceedings of the 8th international natural language generation conference (INLG)*, pages 26–34, 2014.

6 Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. The WebNLG challenge: Generating text from RDF data. In *INLG*, pages 124–133, 2017.

7 Daniel Gerber and A-C Ngonga Ngomo. Bootstrapping the Linked Data Web. In *1st Workshop on Web Scale Knowledge Extraction@ ISWC*, volume 2011, 2011.

8 Nicolas Heist and Heiko Paulheim. Language-Agnostic Relation Extraction from Wikipedia Abstracts. In *The Semantic Web – ISWC 2017*, pages 383–399, 2017.

9 John McCrae, Dennis Spohr, and Philipp Cimiano. Linking Lexical Resources and Ontologies on the Semantic Web with Lemon. In *ESWC*, pages 245–259. Springer, 2011.

10 Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP 2009*, pages 1003–1011, 2009.

11 Diego Moussallem, Dwaraknath Gnaneshwar, Thiago Castro Ferreira, and Axel-Cyrille Ngonga Ngomo. NABU–Multilingual Graph-Based Neural RDF Verbalizer. In *ISWC*, pages 420–437, 2020.

12 Diego Moussallem, René Speck, and Axel-Cyrille Ngonga Ngomo. Generating Explanations in Natural Language from Knowledge Graphs. In *Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges*, volume 47 of *Studies on the Semantic Web*, pages 213–241. IOS Press, 2020.

13 Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. PATTY: A Taxonomy of Relational Patterns with Semantic Types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1135–1145, 2012.

14 Xiantang Sun and Chris Mellish. An experiment on "free generation" from single rdf triples. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*, pages 105–108, 2007.

15 Sebastian Walter, Christina Unger, and Philipp Cimiano. M-ATOLL: A Framework for the Lexicalization of Ontologies in Multiple Languages. In *ISWC*, pages 472–486. Springer, 2014.

16 Tianyi Wu, Yuguo Chen, and Jiawei Han. Re-examination of interestingness measures in pattern mining: a unified framework. *Data Mining and Knowledge Discovery*, 21(3):371–397, 2010.

A Details on and Examples for the Rule Patterns

Rule Patterns $c_s, l_s \Rightarrow po$ and $c_s, ll_s \Rightarrow po$. Given for a rule of type $c_s, l_s \Rightarrow po$ ($c_s, ll_s \Rightarrow po$) are a class $c \in C$, a property $p \in P$, a term $o \in \mathcal{T}$, and a (localized) linguistic pattern l.

$$c \in c_e \wedge l \in l_e \Rightarrow (e, p, o) \in G \qquad (c_s, l_s \Rightarrow po)$$
$$c \in c_e \wedge l \in l_e^{c,p,d} \Rightarrow (e, p, o) \in G \qquad (c_s, ll_s \Rightarrow po)$$

Meaning: Given that in a document that describes an entity e that is an instance of the class c occurs the (localized) linguistic pattern l, the rule predicts that the entity e has the value o for the property p.

Example for rule pattern $c_s, l_s \Rightarrow po$:

$dbo{:}Politician \in c_e \wedge$ "Awami League" $\in l_e$
$\Rightarrow (e, dbo{:}party, dbr{:}Bangladesh\_Awami\_League) \in G$

sup(A)= 40	AllConf(A,B)≈ 0.88
sup(B)= 42	Coherence(A,B)≈ 0.45
sup(AB)= 37	Cosine(A,B)≈ 0.9
P(B|A)≈ 0.88	IR(A,B)≈ 0.04
P(A|B)≈ 0.92	Kulczynski(A,B)≈ 0.9
	MaxConf(A,B)≈ 0.92

Meaning: Given an entity that is an instance of the class $dbo{:}Politician$ and where the document that describes that entity contains the linguistic pattern "Awami League", the rule predicts that the entity is in the relation $dbo{:}party$ with $dbr{:}Bangladesh\_Awami\_League$.

Example for rule pattern $c_s, ll_s \Rightarrow po$:

$dbo{:}Arachnid \in c_e \wedge$
"family Trombidiidae" $\in l_e^{dbo:Arachnid,dbo:genus,so}$
$\Rightarrow (e, dbo{:}genus, dbr{:}Trombidium) \in G$

sup(A)= 40	AllConf(A,B)≈ 0.88
sup(B)= 42	Coherence(A,B)≈ 0.45
sup(AB)= 37	Cosine(A,B)≈ 0.9
P(B|A)≈ 0.88	IR(A,B)≈ 0.04
P(A|B)≈ 0.92	Kulczynski(A,B)≈ 0.9
	MaxConf(A,B)≈ 0.92

Meaning: Given an entity that is an instance of the class $dbo{:}Arachnid$ and where the abstract of that entity contains the localized linguistic pattern "family Trombidiidae" (which is localized to the class $dbo{:}Arachnid$ and the predicate $dbo{:}genus$), the rule predicts that the entity is in the relation $dbo{:}genus$ with $dbr{:}Trombidium$.

Rule Patterns $c_s, l_s \Rightarrow p$ and $c_s, ll_s \Rightarrow p$. Given for a rule of type $c_s, l_s \Rightarrow p$ ($c_s, ll_s \Rightarrow p$) are a class $c \in C$, a property $p \in P$, and a (localized) linguistic pattern l.

$$c \in c_e \wedge l \in l_e \Rightarrow \exists o \in \mathcal{T} : (e, p, o) \in G \qquad (c_s, l_s \Rightarrow p)$$
$$c \in c_e \wedge l \in l_e^{c,p,d} \Rightarrow \exists o \in \mathcal{T} : (e, p, o) \in G \qquad (c_s, ll_s \Rightarrow p)$$

Meaning: Given that in a document that describes an entity e that is an instance of the class c occurs the linguistic pattern l, predict that the entity e has some value for the property p.

Example for rule pattern $c_s, l_s \Rightarrow p$:

$dbo{:}Actor \in c_e \wedge$ "a Swedish actor" $\in l_e$
$\Rightarrow \exists o : (e, dbo{:}nationality, o) \in G$

sup(A)= 213	AllConf(A,B)≈ 0.29
sup(B)= 729	Coherence(A,B)≈ 0.23
sup(AB)= 213	Cosine(A,B)≈ 0.54
P(B|A)≈ 0.29	IR(A,B)≈ 0.71
P(A|B)≈ 1	Kulczynski(A,B)≈ 0.65
	MaxConf(A,B)≈ 1

Meaning: Given an entity that is an instance of the class $dbo{:}Actor$ and where the document that describes that entity contains the linguistic pattern "a Swedish actor", the rule predicts that the entity is in the relation $dbo{:}nationality$ with some entity.

Example for rule pattern $c_s, ll_s \Rightarrow p$:

$dbo{:}Actor \in c_e \wedge$ "married to" $\in l_e^{dbo:Actor,dbo:spouse,so}$	sup(A)= 61 AllConf(A,B)≈ 0.12
$\Rightarrow \exists o : (e, dbo{:}spouse, o) \in G$	sup(B)= 289 Coherence(A,B)≈ 0.1
	sup(AB)= 36 Cosine(A,B)≈ 0.27
	P(B|A)≈ 0.12 IR(A,B)≈ 0.73
	P(A|B)≈ 0.59 Kulczynski(A,B)≈ 0.36
	MaxConf(A,B)≈ 0.59

Meaning: Given an entity that is an instance of the class *dbo*:*Actor* and where the document that describes that entity contains the linguistic pattern "married to" (which is localized to the class *dbo*:*Actor* and the predicate *dbo*:*spouse*), the rule predicts that the entity is in the relation *dbo*:*spouse* with some entity.

Rule Pattern $c_s, l_s \Rightarrow o$. Given for a rule of type $c_s, l_s \Rightarrow o$ are a class $c \in C$, a term $o \in \mathcal{T}$, and a linguistic pattern l.

$$c \in c_e \wedge l \in l_e \Rightarrow \exists p \in \mathcal{U} : (e, p, o) \in G \qquad (c_s, l_s \Rightarrow o)$$

Meaning: Given that in a document that describes an entity e that is an instance of the class c occurs the linguistic pattern l, predict that there is some relation by which e is related to the term o.

Example for rule pattern $c_s, l_s \Rightarrow o$:

$dbo{:}Grape \in c_e \wedge$ "white" $\in l_e$	sup(A)= 225 AllConf(A,B)≈ 0.81
$\Rightarrow \exists p : (e, p,$ "Blanc"@$en) \in$	sup(B)= 198 Coherence(A,B)≈ 0.43
	sup(AB)= 183 Cosine(A,B)≈ 0.87
	P(B|A)≈ 0.92 IR(A,B)≈ 0.11
	P(A|B)≈ 0.81 Kulczynski(A,B)≈ 0.87
	MaxConf(A,B)≈ 0.92

Meaning: Given an entity that is an instance of the class *dbo*:*Grape* and where the document that describes that entity contains the linguistic pattern "white", the rule predicts that the entity is in some relation with the term "Blanc"@*en*.

Rule Patterns $c_o, l_o \Rightarrow sp$ **and** $c_o, ll_o \Rightarrow sp$. Given for a rule of type $c_o, l_o \Rightarrow sp$ ($c_o, ll_o \Rightarrow sp$) are a class $c \in C$, a term $s \in \mathcal{U} \cup \mathcal{B}$, a predicate $p \in \mathcal{P}$, and a (localized) linguistic pattern l.

$$c \in c_e \wedge l \in l_e \Rightarrow (s, p, e) \in G \qquad (c_o, l_o \Rightarrow sp)$$
$$c \in c_e \wedge l \in l_e^{c,p,d} \Rightarrow (s, p, e) \in G \qquad (c_o, ll_o \Rightarrow sp)$$

Meaning: Given that in a document that describes an entity e that is an instance of the class c occurs the (localized) linguistic pattern l, predict that there is an entity s that is in relation p with the entity e.

Example for rule pattern $c_o, l_o \Rightarrow sp$:

$dbo{:}Island \in c_e \wedge$ "Baltic" $\in l_e$	sup(A)= 43 AllConf(A,B)≈ 0.35
$\Rightarrow (dbr{:}Baltic\_Sea, dbo{:}island, e) \in G$	sup(B)= 23 Coherence(A,B)≈ 0.23
	sup(AB)= 15 Cosine(A,B)≈ 0.48
	P(B|A)≈ 0.65 IR(A,B)≈ 0.39
	P(A|B)≈ 0.35 Kulczynski(A,B)≈ 0.5
	MaxConf(A,B)≈ 0.65

Meaning: Given an entity that is an instance of the class *dbo*:*Island* and where the document that describes that entity contains the linguistic pattern "Baltic", the rule predicts that the entity *dbr*:*Baltic_See* is in the relation *dbo*:*island* with this entity.

Example for rule pattern $c_o, ll_o \Rightarrow sp$:

$dbo{:}Artwork \in c_e \wedge$ "Salvador" $\in l_e^{dbo:Artwork,dbo:notableWork,so}$	sup(A)= 6 AllConf(A,B)≈ 0.86
$\Rightarrow (dbr{:}Salvador\_Dalí, dbo{:}notableWork, e) \in G$	sup(B)= 7 Coherence(A,B)≈ 0.46
	sup(AB)= 6 Cosine(A,B)≈ 0.93
	P(B|A)≈ 0.86 IR(A,B)≈ 0.14
	P(A|B)≈ 1 Kulczynski(A,B)≈ 0.93
	MaxConf(A,B)≈ 1

Meaning: Given an entity that is an instance of the class *dbo*:*Artwork* and where the document that describes that entity contains the linguistic pattern "Salvador" (which is localized to the class *dbo*:*Artwork* and the predicate *dbo*:*notableWork*), the rule predicts that the entity *dbr*:*Salvador_Dalí* is in the relation *dbo*:*notableWork* with this entity.

Rule Pattern $c_o, l_o \Rightarrow s$. Given for a rule of type $c_o, l_o \Rightarrow s$ are a class $c \in C$, a term $o \in \mathcal{T}$, and a linguistic pattern l.

$$c \in c_e \wedge l \in l_e \Rightarrow (s, p, e) \in G \qquad (c_o, l_o \Rightarrow s)$$

Meaning: Given that in a document that describes an entity e that is an instance of the class c occurs the linguistic pattern l, predict that there is an entity s that is in some relation with the entity e.

Example for rule pattern $c_o, l_o \Rightarrow s$:

dbo:*Language* $\in c_e \wedge$ "Nahuatl" $\in l_e$	sup(A)= 21	AllConf(A,B)≈ 0.76
	sup(B)= 18	Coherence(A,B)≈ 0.41
$\Rightarrow \exists p : (dbr{:}Nahuan\_languages, p, e) \in G$	sup(AB)= 16	Cosine(A,B)≈ 0.82
	P(B|A)≈ 0.89	IR(A,B)≈ 0.13
	P(A|B)≈ 0.76	Kulczynski(A,B)≈ 0.83
		MaxConf(A,B)≈ 0.89

Meaning: Given an entity that is an instance of the class *dbo*:*Language* and where the document that describes that entity contains the linguistic pattern "Nahuatl", the rule predicts that the entity *dbr*:*Nahuan_languages* is in some relation with this entity.

Rule Patterns $c_o, l_o \Rightarrow p$ **and** $c_o, ll_o \Rightarrow p$. Given for a rule of type $c_o, l_o \Rightarrow p$ ($c_o, ll_o \Rightarrow p$) are a class $c \in C$, a predicate $p \in \mathcal{P}$, and a (localized) linguistic pattern l.

$$c \in c_e \wedge l \in l_e \Rightarrow \exists s \in \mathcal{U} \cup \mathcal{B} : (s, p, e) \in G \qquad (c_o, l_o \Rightarrow p)$$
$$c \in c_e \wedge l \in l_e^{c,p,d} \Rightarrow \exists s \in \mathcal{U} \cup \mathcal{B} : (s, p, e) \in G \qquad (c_o, ll_o \Rightarrow p)$$

Meaning: Given that in a document that describes an entity e that is an instance of the class c occurs the (localized) linguistic pattern l, predict that there is some entity that is in the relation p with the entity e.

Example for rule pattern $c_o, l_o \Rightarrow p$:

dbo:*AmateurBoxer* $\in c_e \wedge$ "silver medal" $\in l_e$	sup(A)= 70	AllConf(A,B)≈ 0.31
	sup(B)= 29	Coherence(A,B)≈ 0.22
$\Rightarrow \exists s : (s, dbo{:}silverMedalist, e) \in G$	sup(AB)= 22	Cosine(A,B)≈ 0.49
	P(B|A)≈ 0.76	IR(A,B)≈ 0.53
	P(A|B)≈ 0.31	Kulczynski(A,B)≈ 0.54
		MaxConf(A,B)≈ 0.76

Meaning: Given an entity e that is an instance of the class *dbo*:*AmateurBoxer* and where the document that describes that entity contains the linguistic pattern "silver medal", the rule predicts that there is some entity which is related via the relation *dbo*:*silverMedalist* to the entity e.

Example for rule pattern $c_o, ll_o \Rightarrow p$:

dbo:*Noble* $\in c_e \wedge$ "married" $\in l_e^{dbo:Noble,dbo:spouse,so}$	sup(A)= 220	AllConf(A,B)≈ 0.1
	sup(B)= 1588	Coherence(A,B)≈ 0.08
$\Rightarrow \exists s : (s, dbo{:}spouse, e) \in G$	sup(AB)= 151	Cosine(A,B)≈ 0.26
	P(B|A)≈ 0.1	IR(A,B)≈ 0.83
	P(A|B)≈ 0.69	Kulczynski(A,B)≈ 0.39
		MaxConf(A,B)≈ 0.69

Meaning: Given an entity e that is an instance of the class *dbo*:*Noble* and where the document that describes that entity contains the linguistic pattern "married" (which is localized to the class *dbo*:*Noble* and the predicate *dbo*:*spouse*), the rule predicts that there is some entity which is related via the relation *dbo*:*spouse* to the entity e.

Rule Patterns $c_s, po \Rightarrow l_s$ and $c_s, po \Rightarrow ll_s$. Given for a rule of type $c_s, po \Rightarrow l_s$ ($c_s, po \Rightarrow ll_s$) are a class $c \in C$, a predicate $p \in \mathcal{P}$, a term $o \in \mathcal{T}$, and a (localized) linguistic pattern l.

$$c \in c_e \wedge (e, p, o) \in G \Rightarrow l \in l_e \qquad (c_s, po \Rightarrow l_s)$$
$$c \in c_e \wedge (e, p, o) \in G \Rightarrow l \in l_e^{c,p,d} \qquad (c_s, po \Rightarrow ll_s)$$

Meaning: Given an entity e that is an instance of the class c and given that e is in relation p to the term o, predict that the text that describes e contains the (localized) linguistic pattern l.

Example for rule pattern $c_s, po \Rightarrow l_s$:

$dbo{:}Actor \in c_e \wedge (e, dbo{:}nationality, dbr{:}Sweden) \in G$	sup(A)= 589	AllConf(A,B)≈ 0.98
$\Rightarrow$ "Swedish" $\in l_e$	sup(B)= 582	Coherence(A,B)≈ 0.49
	sup(AB)= 579	Cosine(A,B)≈ 0.99
	P(B|A)≈ 0.99	IR(A,B)≈ 0.01
	P(A|B)≈ 0.98	Kulczynski(A,B)≈ 0.99
		MaxConf(A,B)≈ 0.99

Meaning: Given an entity e that is an instance of the class *dbo*:*Actor* and where the entity is in the relation *dbo*:*nationality* with the entity *dbr*:*Sweden*, the rule predicts that the linguistic pattern "Swedish" occurs in the text about the entity e.

Example for rule pattern $c_s, po \Rightarrow ll_s$:

$dbo{:}Criminal \in c_e \wedge (e, dbo{:}deathPlace, dbr{:}Sicily) \in G$	sup(A)= 11	AllConf(A,B)≈ 0.64
$\Rightarrow$ "Mafia" $\in l_e^{dbo:Criminal, dbo:deathPlace, os}$	sup(B)= 8	Coherence(A,B)≈ 0.37
	sup(AB)= 7	Cosine(A,B)≈ 0.75
	P(B|A)≈ 0.88	IR(A,B)≈ 0.25
	P(A|B)≈ 0.64	Kulczynski(A,B)≈ 0.76
		MaxConf(A,B)≈ 0.88

Meaning: Given an entity e that is an instance of the class *dbo*:*Criminal* and where the entity is in the relation *dbo*:*deathPlace* with the entity *dbr*:*Sicily*, the rule predicts that the localized linguistic pattern "Mafia" (which is localized to the class *dbo*:*Criminal* and the predicate *dbo*:*deathPlace*) occurs in the text about the entity e.

Rule Patterns $c_s, p \Rightarrow l_s$ and $c_s, p \Rightarrow ll_s$. Given for a rule of type $c_s, p \Rightarrow l_s$ ($c_s, p \Rightarrow l_s$) are a class $c \in C$, a predicate $p \in \mathcal{P}$, and a (localized) linguistic pattern l.

$$c \in c_e \wedge \exists o \in \mathcal{T} : (e, p, o) \in G \Rightarrow l \in l_e \qquad (c_s, p \Rightarrow l_s)$$
$$c \in c_e \wedge \exists o \in \mathcal{T} : (e, p, o) \in G \Rightarrow l \in l_e^{c,p,d} \qquad (c_s, p \Rightarrow ll_s)$$

Meaning: Given an entity e that is an instance of the class c and given that e is in relation p to some term, predict that the text that describes e contains the (localized) linguistic pattern l.

Example for rule pattern $c_s, p \Rightarrow l_s$:

$dbo{:}Fungus \in c_e \wedge \exists o : (e, dbp{:}genusAuthority, o) \in G$	sup(A)= 4330	AllConf(A,B)≈ 0.86
$\Rightarrow$ "is a genus" $\in l_e$	sup(B)= 3773	Coherence(A,B)≈ 0.46
	sup(AB)= 3717	Cosine(A,B)≈ 0.92
	P(B|A)≈ 0.99	IR(A,B)≈ 0.13
	P(A|B)≈ 0.86	Kulczynski(A,B)≈ 0.92
		MaxConf(A,B)≈ 0.99

Meaning: Given an entity e that is an instance of the class *dbo*:*Fungus* and where the entity is in the relation *dbo*:*genusAuthority* with some term, the rule predicts that the linguistic pattern "is a genus" occurs in the text about the entity e.

Example for rule pattern $c_s, p \Rightarrow ll_s$:

$dbo{:}CricketGround \in c_e \wedge \exists o : (e, dbp{:}location, o) \in G$	sup(A)= 195	AllConf(A,B)≈ 0.33
$\Rightarrow$ "is a cricket ground in" $\in l_e^{dbo:CricketGround, dbp:location, so}$	sup(B)= 64	Coherence(A,B)≈ 0.25
	sup(AB)= 64	Cosine(A,B)≈ 0.57
	P(B|A)≈ 1	IR(A,B)≈ 0.67
	P(A|B)≈ 0.33	Kulczynski(A,B)≈ 0.66
		MaxConf(A,B)≈ 1

Meaning: Given an entity e that is an instance of the class *dbo*:*CricketGround* and where the entity is in the relation *dbp*:*location* with some term, the rule predicts that the localized linguistic pattern "is a cricket ground in" (which is localized to the class *dbo*:*CricketGround* and the predicate *dbp*:*location*) occurs in the text about the entity e.

Rule Pattern $c_s, o \Rightarrow l_s$. Given for a rule of type $c_s, o \Rightarrow l_s$ are a class $c \in C$, a term $o \in \mathcal{T}$, and a linguistic pattern l.

$$c \in c_e \wedge \exists p \in \mathcal{U} : (e, p, o) \in G \Rightarrow l \in l_e \qquad (c_s, o \Rightarrow l_s)$$

Meaning: Given an entity e that is an instance of the class c and given that e is in some relation to the term o, predict that the text that describes e contains the (localized) linguistic pattern l.

Example for rule pattern $c_s, o \Rightarrow l_s$:

$dbo{:}Protein \in c_e \wedge \exists p : (e, p, "MT") \in G$	sup(A)= 24	AllConf(A,B)≈ 1
$\Rightarrow$ "Mitochondrially encoded" $\in l_e$	sup(B)= 24	Coherence(A,B)≈ 0.5
	sup(AB)= 24	Cosine(A,B)≈ 1
	P(B|A)≈ 1	IR(A,B)≈ 0
	P(A|B)≈ 1	Kulczynski(A,B)≈ 1
		MaxConf(A,B)≈ 1

Meaning: Given an entity e that is an instance of the class *dbo*:*Protein* and where the entity is in some relation with the term "*MT*", the rule predicts that the linguistic pattern "Mitochondrially encoded" occurs in the text about the entity e.

Rule Patterns $c_o, sp \Rightarrow l_o$ **and** $c_o, sp \Rightarrow ll_o$. Given for a rule of type $c_o, sp \Rightarrow l_o$ ($c_o, sp \Rightarrow ll_o$) are a class $c \in C$, a term $s \in \mathcal{U} \cup \mathcal{B}$, a predicate $p \in \mathcal{P}$, and a (localized) linguistic pattern l.

$$c \in c_e \wedge (s, p, e) \in G\} \Rightarrow l \in l_e \qquad (c_o, sp \Rightarrow l_o)$$
$$c \in c_e \wedge (s, p, e) \in G\} \Rightarrow l \in l_e^{c,p,d} \qquad (c_o, sp \Rightarrow ll_o)$$

Meaning: Given an entity e that is an instance of the class c and given that the term s is in relation p with e, predict that the text that describes e contains the (localized) linguistic pattern l.

Example for rule pattern $c_o, sp \Rightarrow l_o$:

$dbo{:}WineRegion \in c_e \wedge (dbr{:}Mendocino\_County\_wine,$	sup(A)= 11	AllConf(A,B)≈ 0.92
$dbp{:}subRegions, e) \in G$	sup(B)= 12	Coherence(A,B)≈ 0.48
$\Rightarrow$ "Mendocino County California $\in l_e$	sup(AB)= 11	Cosine(A,B)≈ 0.96
	P(B|A)≈ 0.92	IR(A,B)≈ 0.08
	P(A|B)≈ 1	Kulczynski(A,B)≈ 0.96
		MaxConf(A,B)≈ 1

Meaning: Given an entity e that is an instance of the class *dbo*:$WineRegion$ and where the entity *dbr*:$Mendocino\_County\_wine$ is in the relation *dbp*:*subRegions* with e, the rule predicts that the linguistic pattern "Mendocino County California" occurs in the text about the entity e.

Example for rule pattern $c_o, sp \Rightarrow ll_o$:

$dbo{:}Airline \in c_e \wedge (dbr{:}Lufthansa, dbo{:}subsidiary, e) \in G$	sup(A)= 11	AllConf(A,B)≈ 0.09
$\Rightarrow$ "subsidiary of" $\in l_e^{dbo:Airline, dbo:subsidiary, so}$	sup(B)= 64	Coherence(A,B)≈ 0.08
	sup(AB)= 6	Cosine(A,B)≈ 0.23
	P(B|A)≈ 0.09	IR(A,B)≈ 0.77
	P(A|B)≈ 0.55	Kulczynski(A,B)≈ 0.32
		MaxConf(A,B)≈ 0.55

Meaning: Given an entity e that is an instance of the class *dbo*:*Airline* and where the entity *dbr*:$Lufthansa$ is in the relation *dbo*:*subsidiary* with e, the rule predicts that the localized linguistic pattern "subsidiary of" (which is localized to the class *dbo*:*Airlines* and the predicate *dbo*:*subsidiary*) occurs in the text about the entity e.

Rule Pattern $c_o, s \Rightarrow l_o$. Given for a rule of type $c_o, s \Rightarrow l_o$ are a class $c \in C$, a term $s \in \mathcal{U} \cup \mathcal{B}$, and a linguistic pattern l.

$$c \in c_e \wedge \exists p \in \mathcal{U} : (s, p, e) \in G\} \Rightarrow l \in l_e \qquad (c_o, s \Rightarrow l_o)$$

Meaning: Given an entity e that is an instance of the class c and given that the term s is in some with e, predict that the text that describes e contains the (localized) linguistic pattern l.

Example for rule pattern $c_o, s \Rightarrow l_o$:

dbo:$Horse \in c_e \wedge \exists p : ($*dbr*:$Orme\_(horse), p, e) \in G$	sup(A)= 14	AllConf(A,B)≈ 0.43
$\Rightarrow$ "English Thoroughbred racehorse" $\in l_e$	sup(B)= 11	Coherence(A,B)≈ 0.24
	sup(AB)= 6	Cosine(A,B)≈ 0.48
	P(B|A)≈ 0.55	IR(A,B)≈ 0.16
	P(A|B)≈ 0.43	Kulczynski(A,B)≈ 0.49
		MaxConf(A,B)≈ 0.55

Meaning: Given an entity e that is an instance of the class *dbo*:$Horse$ and where the entity *dbr*:$Orme\_(horse)$ is in some relation with e, the rule predicts that the linguistic pattern "English Thoroughbred racehorse" occurs in the text about the entity e.

Rule Patterns $c_o, p \Rightarrow l_o$ **and** $c_o, p \Rightarrow ll_o$. Given for a rule of type $c_o, p \Rightarrow l_o$ $(c_o, p \Rightarrow ll_o)$ are a class $c \in C$, a predicate $p \in \mathcal{P}$, and a (localized) linguistic pattern l.

$$c \in c_e \wedge \exists s \in \mathcal{U} \cup \mathcal{B} : (s, p, e) \in G \Rightarrow l \in l_e \qquad (c_o, p \Rightarrow l_o)$$
$$c \in c_e \wedge \exists s \in \mathcal{U} \cup \mathcal{B} : (s, p, e) \in G \Rightarrow l \in l_e^{c,p,d} \qquad (c_o, p \Rightarrow ll_o)$$

Meaning: Given an entity e that is an instance of the class c and given that some term is in relation p with e, predict that the text that describes e contains the (localized) linguistic pattern l.

Example for rule pattern $c_o, p \Rightarrow l_o$:

dbo:$Wrestler \in c_e \wedge \exists s : (s,$ *dbp*:$bronze, e) \in G$	sup(A)= 30	AllConf(A,B)≈ 0.47
$\Rightarrow$ "bronze medal" $\in l_e$	sup(B)= 29	Coherence(A,B)≈ 0.24
	sup(AB)= 14	Cosine(A,B)≈ 0.47
	P(B|A)≈ 0.48	IR(A,B)≈ 0.02
	P(A|B)≈ 0.47	Kulczynski(A,B)≈ 0.47
		MaxConf(A,B)≈ 0.48

Meaning: Given an entity e that is an instance of the class *dbo*:$Wrestler$ and where some entity is in the relation *dbp*:*bronze* with e, the rule predicts that the linguistic pattern "bronze medal" occurs in the text about the entity e.

Example for rule pattern $c_o, p \Rightarrow ll_o$:

dbo:$Crustacean \in c_e \wedge \exists s : (s,$ *dbp*:$superfamilia, e) \in G$	sup(A)= 33	AllConf(A,B)≈ 0.42
$\Rightarrow$ "is a superfamily" $\in l_e^{dbo:Crustacean,dbp:superfamilia,so}$	sup(B)= 15	Coherence(A,B)≈ 0.29
	sup(AB)= 14	Cosine(A,B)≈ 0.63
	P(B|A)≈ 0.93	IR(A,B)≈ 0.53
	P(A|B)≈ 0.42	Kulczynski(A,B)≈ 0.68
		MaxConf(A,B)≈ 0.93

Meaning: Given an entity e that is an instance of the class *dbo*:$Crustacean$ and where some entity is in the relation *dbp*:$superfamilia$ with e, the rule predicts that the localized linguistic pattern "is a superfamily" (which is localized to the class *dbo*:$Crustacean$ and the predicate *dbo*:$superfamilia$) occurs in the text about the entity e.