Can Predicate Lexicalizations Help in Named Entity Disambiguation?

Position Paper

Heiko Paulheim[1] and Christina Unger[2]

University of Mannheim, Germany
Data and Web Science Group
{heiko}@informatik.uni-mannheim.de and Semantic Computing Group, CITEC,
Bielefeld University
cunger@techfak.uni-bielefeld.de

Abstract. Most named entity disambiguation approaches use various resources, such as surface form catalogues and relations of entities in the target knowledge base. In contrast, predicates that describe relations between the entity mentions in text are only scarcely exploited. In this position paper, we argue how predicates, i.e., surface forms for relations in the target knowledge base, can potentially help to improve named entity disambiguation results.

Keywords: Named Entity Disambiguation, Ontology Lexicon, Knowledge Base Lexicalization, DBpedia

1 Motivation

The identification of entities in text usually comprises two steps. First, mentions of entities are *recognized*, which often involves a big amount of ambiguity. For example, the expression *Heidi* could refer to the model Heidi Klum, the Swiss children book, and so on. Therefore, the recognized mentions need to be *disambiguated*. This second step is often called *named entity disambiguation* (NED) or *entity linking*, as it involves linking mentions to unique identifiers in a knowledge base. For example, the entity mention *Heidi* in a sentence such as *Heidi and her husband Seal live in Vegas* would be linked to the DBpedia [2] resource `dbr:Heidi_Klum`, while the same mention in a sentence such as *Heidi was written by Swiss author Johanna Spyri* would be linked to the DBpedia resource `dbr:Heidi`, representing the children book.

Named entity disambiguation often uses dictionaries which collect textual surface forms of entities, e.g. mapping the forms *New York*, *NCY*, and *Big Apple* to the DBpedia entity `dbr:New_York_City`. In many cases, also co-occurences and relations between entities are taken into account for disambiguation. For example, in the sentence *Cairo was the code name for a project at Microsoft from 1991 to 1996*, the co-occurence of *Cairo* with *Microsoft* allows to link it to the operating system instead of the Egypt city. However, co-occuring entities are not always sufficient for disambiguation. For example, in the sentence *While Apple is an electronics company, Mango is a clothing one*, the co-occurence of *Apple* and *Mango* does not provide enough context to distinguish between companies and fruits.

To the best of our knowledge, NED approaches usually do not exploit predicates occuring in texts along with entities, such as *husband* or *written by* in the Heidi example, or *company* in the Apple and Mango example. In this paper, we argue that such predicates are actually a helpful source of knowledge to improve NED, especially when little other context is available for disambiguation. We demonstrate this using examples from the KORE50 benchmark [1], exploiting property lexicalizations. For example, for the property `spouse`, typical lexicalizations are *married to*, *husband of*, and *wife of*.

Such lexicalizations can help named entity disambiguation in two respects. First, properties in knowledge bases such as DBpedia often specify domain and range information in their ontologies, i.e., valid classes of entities that can appear in the subject and object position of a statement using that property. This domain and range information can be used to discard NED candidates that are inconsistent with the ontology. For example, consider the following KORE50 sentence:

> *David and Victoria named their children Brooklyn, Romeo, Cruz, and Harper Seven.*

Here, *Brooklyn* is easily confused with the New York City borough `Brooklyn` by NED tools. However, taking the predicate *children* into account, which is a lexicalization of the property `child`, we can discard this misleading option because the domain and range of `child` are persons, while `Brooklyn` is a place.

The second possible use of property lexicalizations is that we can explicitly search for relations between entities in the knowledge base. For example, in the above case, we would already have learned that the mentioned entities (*Brooklyn*, *Romeo*, etc.) are persons. Given that we already correctly disambiguated one of the entities, we can use this information to search for entities that stand in the `child` relation to it. For example, if we already linked *David* to `David_Beckham`, we can use the DBpedia triple

```
dbr:David_Beckham dbo:child dbr:Brooklyn_Beckham .
```

to link *Brooklyn* to `dbr:Brooklyn_Beckham`, instead of any other person.

In order to exploit such information, lexicalizations of properties are required. One such collection is *DBlexipedia*.

2 Predicate Lexicalizations in DBlexipedia

DBlexipedia [7] is an ontology lexicon that connects properties in the DBpedia ontology to common surface forms that express them in a particular natural language, together with linguistic information about their morpho-syntactic properties.

The lexicon published on `http://dblexipedia.org` is the result of applying the automatic ontology lexicon induction method M-ATOLL [5, 6], which creates ontology lexica in *lemon* [3] format as follows. It takes as input an ontology and dataset (here, DBpedia) and a dependency parsed text corpus in the target language (here, English Wikipedia). As first step, M-ATOLL retrieves all triples for a given property from the dataset. For example, the results for the property `spouse` include the triple

`<Lulu,spouse,Maurice_Gibb>`. Then, it retrieves all sentences from the parsed text corpus that contain mentions of the subject and object of the extracted triples, e.g. *In 1969 the singer Lulu married Maurice Gibb*. It searches for predefined patterns in those sentences, in order to extract candidate lexicalizations of the property, such as *to marry*.

So far, M-ATOLL covers entries that describe transitive verbs (e.g. *to cross*), intransitive verbs with a prepositional object (e.g. *to live in*), relational nouns with prepositional object (e.g. *capital of*), and relational adjectives (e.g. *similar to*).

3 Preliminary Experiment

To analyze the potential value of property lexicalizations for the NED task, we analyzed the 50 sentences of the KORE50 corpus. We processed each of those sentences using DBpedia Spotlight [4] in the standard settings. Out of the 50 cases, DBpedia Spotlight performed a wrong disambiguation for at least one entity in 37 cases.

Next, we looked at the errors made, and analyzed whether the error could potentially be solved by using information on predicates occuring in the sentence. To that end, we looked up the predicate in DBlexipedia. If we found it as lexicalization of a DBpedia property, we marked the error as potentially solveable if

- a wrongly disambiguated entity had a type which was inconsistent with the respective property's domain or range, or
- a wrongly disambiguated entity had a direct connection through the found property to the correct entity.

For example, In the following KORE50 sentence, DBpedia Spotlight correctly links *Angelina* to `Angelina_Jolie`, but fails to link *Jon* and *Brad* to the correct entities.

> *Angelina, her father Jon, and her partner Brad never played together in the same movie.*

The predicate *father*, however, can be found in DBlexipedia as lexialization of the property `child`, which links `Angeline_Jolie` to `Jon_Voight`, the correct linking for *Jon*.

Similarly, in the following sentence, both *Hurricane* and *Desire* are incorrectly linked by DBpedia Spotlight.

> *Dylan performed Hurricane about the black fighter Carter, from his album Desire.*

Here, the predicate *perform* is found as lexicalization of the property `musicalArtist` with domain `Single` and range `MusicalArtist`, which helps disambiguating *Hurricane* to the Bob Dylan single. Furthermore, *album* is found as lexicalization of the property `artist`, which relates `Bob_Dylan` with his album `Desire`.

Also the mention *John* in the following sentence is incorrectly linked.

> *Pixar produced Cars, and John directed it.*

But it could be correclty linked using the lexical knowledge from DBlexipedia that *direct* expresses the property `director`, and the factual knowledge from DBpedia that the correctly identified movie `Cars` stands in the `director` relation to the entity `John_Lasseter`.

In total, in 17 out of the 37 cases where DBpedia Spotlight performed a wrong disambiguation, the error could have been identified with either one of the two strategies.[1] These 17 cases are distributed quite equally across domains:

- CEL (Celebrities): 4/8
- MUS (Music): 4/8
- BUS (Business): 1/8
- SPO (Sports): 4/7
- POL (Politics): 4/6

In addition, we can identify cases where the proposed approach cannot help. First, it can happen that a predicate is not contained in DBlexipedia. For example, neither *drop out* nor *join* are listed as lexicalizations of any property, so they cannot be used for disambiguating *Steve* in *Steve dropped out of Stanford to join Microsoft*.

Second, it can happen that a lexicalization is found but either does not point to the correct property, or the corresponding triple in DBpedia is missing. For example, in the phrase *Steve, the former CEO of Apple*, DBlexipedia does list *CEO of* as lexicalization of the property `keyPerson`, but in DBpedia `Steve_Jobs` is related to `Apple_Inc.` by means of `board` and `occupation`.

Third, there are sentences without an explicit predicate between entity mentions, as the following one:

> *Steve, Bill, Sergey, and Larry have drawn a great deal of admiration these days for their pioneering successes that changed the world we live in.*

Analogously, there are sentences that contain predicates but the expressed relation is not modelled in DBpedia. For example, the sentence *Müller scored a hattrick against England* contains the predicate *score against*, which does not correspond to any property in DBpedia. Similar cases affect predicates that are modeled through more complex constructs, such as property paths or reifications.

4 Conclusion

This preliminary experiment shows that predicates, i.e. natural language lexicalizations of properties in the knowledge base, are a valuable source of knowledge when trying to improve the results of NED in cases where only little context is available for disambiguation. Although a formal evaluation on an actual implementation is still missing, the findings from the experiments are quite promising.

[1] However, there may be more than one error in the sentences, and in some cases, we would not be able to address all of those. Hence, this should not be misread as "half of the errors can be identified."

References

1. Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 545–554. ACM, 2012.
2. Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 2013.
3. John McCrae, Dennis Spohr, and Philipp Cimiano. Linking lexical resources and ontologies on the semantic web with lemon. In *The Semantic Web: Research and Applications*, pages 245–259. Springer, 2011.
4. Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, 2011.
5. Sebastian Walter, Christina Unger, and Philipp Cimiano. ATOLL – A framework for the automatic induction of ontology lexica. *Data & Knowledge Engineering*, 94, Part B(0):148–162, 2014. Special Issue following the 18th International Conference on Applications of Natural Language Processing to Information Systems (NLDB'13).
6. Sebastian Walter, Christina Unger, and Philipp Cimiano. M-ATOLL: A framework for the lexicalization of ontologies in multiple languages. In *The Semantic Web – ISWC 2014*, volume 8796 of *Lecture Notes in Computer Science*, pages 472–486. Springer, 2014.
7. Sebastian Walter, Christina Unger, and Philipp Cimiano. Dblexipedia: A nucleus for a multilingual lexical semantic web. In *3rd International Workshop on NLP&DBpedia*, 2015.