# Named Entity Disambiguation using Linked Data

Danica Damljanovic and Kalina Bontcheva

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield, UK
d.damljanovic, k.bontcheva@dcs.shef.ac.uk

## 1 Introduction

Identification of Named Entities (NE) such as people, organisations and locations is fundamental to semantic annotation and is the starting point of more advanced text mining algorithms. For instance, sentiment analysis is widely used in finance to extract the latest signals and events from news that could affect stock prices. However, before extracting company-related sentiment, it is necessary to identify the documents containing the corresponding and *unambiguous* company entities. Humans usually resolve ambiguities based on context. We argue that Linked Data can be a valuable source for extending the already available context. We combine a state-of-the-art named entity tool with novel Linked Data-based similarity measures and show that our algorithm can improve disambiguation accuracy on a subset of Wikipedia user profiles.

## 2 Entity Linking and Disambiguation Algorithm

The goal of the algorithm is to identify named entities in text and attach the correct DBpedia URI to each one of them. For the former, we use the ANNIE Information Extraction system from GATE [1]. It combines some small lists of names (e.g. days of the week, months) and rule-based grammars, to processes text and produce NE types such as *Organization*, *Location* and *Person*. ANNIE also resolves coreference so that entities with the same meaning are linked. For example, *General Motors* and *GM* would be identified as referring to the same entity.

GATE's ontology-based gazetteer, namely the Large Knowledge Gazetteer (LKB), is used for entity linking. LKB performs lookup and assigns URIs to words/phrases in the text. For the purpose of our application, we match only against the values of the *rdf:label* and *foaf:name* properties, for all instances of the *dbpedia-ont:Person*, *dbpedia-ont:Organisation* and *dbpedia-ont:Place* classes.

Both ANNIE and LKB can be used independently, however, while NE types generated by ANNIE miss the URI which is necessary to disambiguate them, LKB does not use any context, which results in generating many spurious entities. For example, each letter *B* is annotated as a possible mention of *dbpedia:B_%28Los_Angeles_Railway%29*, which refers to a line called *B* operated by Los Angeles Railway. We next describe the algorithm which filters out such noise, by consolidating the output of ANNIE and LKB, followed by a disambiguation step. A high-level pseudo code looks as follows:

```
1. Identify NEs (Location, Organisation and Person) using ANNIE
2. For each NE add URIs of matching instances from DBpedia
3. For each ambiguous NE calculate disambiguation scores
4. Remove all matches except the highest scoring one
```

The disambiguation algorithm uses context in which the particular entity appears and a weighted sum of the following three similarity metrics:

- *String similarity*: refers to the Levenshtein distance between the text string (such as *Paris*), and the labels describing the entity URIs (for example, *Paris Hilton*, *Paris* and *Paris, Ontario*).
- *Structural similarity* is calculated based on whether the ambiguous NE has a relation with any other NE from the same sentence or document. For example, if the document mentions both *Paris* and *France*, then structural similarity indicates that *Paris* refers to the capital of France. All other entity URIs can be disregarded, based on the existing relationship between *dbpedia:Paris* and *dbpedia:France*.
- *Contextual similarity* is calculated based on the probability that two words have a similar meaning as in a large corpus (DBpedia abstracts in our case) they appear with a similar set of other words. To implement that we use the Random Indexing method [2] and calculate similarity using the cosine function.

## 3 Experiments

We manually labelled the corpus with 100 Wikipedia user profiles to create a gold standard against which we can evaluate performance, using precision, recall and f-measure. Table 1 summarises the results. The addition of ANNIE to the DBpedia lookup using

| | Precision | Recall | F-measure |
|---|---|---|---|
| LKB | 0.03 | 0.86 | 0.05 |
| LKB+ANNIE | 0.14 | 0.81 | 0.24 |
| LKB+ANNIE+Disambiguation | 0.84 | 0.80 | 0.82 |

**Table 1.** Precision, recall and f-measure of the different algorithms

LKB improved precision at a slight cost in recall. Adding the disambiguation layer results in further improved precision with slightly lower recall. However this results in an overall improvement in f-measure, which rises to 0.82. This shows that our disambiguation algorithm which exploits Linked Data as an additional knowledge source, eliminated a large number of incorrect annotations.

**Acknowledgements** This research has been supported by the EU-funded FP7 TrendMiner[1] project.

[1] http://www.trendminer-project.eu/