Where is the Context for Disambiguation? An Experimental Analysis.

Pablo N. Mendes

August 31, 2013

Abstract

1 Introduction

This papers presents an experimental analysis of the impact of importing context information from different ontologies such as DBpedia in the recognition and semantic disambiguation of concepts and entities extracted from well known knowledge bases. We strongly believe that some type of context of certain phrases support a better disambiguation. Through this paper we aim to provide an efficient way to extract context in order to improve the disambiguation of named entities. We show that the process of indexing context (...)

The DBpedia Spotlight process the linking of unstructured information sources to the Linked Open Data cloud through DBpedia, by automatically annotating mentions of DBpedia resources. It is able to recognize(...). Is also offers many potential sources of context to model each entity, such as title, sentence, paragraph, content, anchor, and so on. Through the contextual information extraction process for the disambiguation process, as well as the indexing of named entities among candidates entities of unstructured texts, the experimental analysis shows a higher disambiguation accuracy in comparison with other research on Wikipedia resource.

DBpedia Spotlight relies on a number of data sources in order to support annotation tasks. Some sources provide metadata such as the types of entity: Person, Location or Organization, for instance. Other sources provide statistical context in which certain words are more strongly associated with certain entities. Thus, this analysis will allow us to evaluate which sources of information provide best information for disambiguation.

We have many potential sources of context to model each DBpedia entity: -title (the title of the article associated with concept c), -sentence (the first sentence of the article), -paragraph (the first paragraph of the article), -content (the full contents of the article), and -anchor (the aggregated anchor texts of all incoming links in Wikipedia). -aggregation of all paragraphs like we do in spotlight, -text from ontology – all properties – all objects – etc.

hypotheses

When modeling entity context for disambiguation, - entity names are the most important part of the text, therefore we can safely discard other words -

there is a trade-off when discarding words. Discarding low frequency words does not hurt performance - there is a trade-off when discarding words. Discarding high frequency words does not hurt performance - (can we test if words are distributionally independent from entity names) - words provide context that entity names cannot capture

When looking at co-occurring entity names for modeling context, - entities that occur in the entity's article are better descriptors - entities that occur in the first section of the entity's article are better descriptors - entities that occur in text outside of entity's articles are better representatives of occurrences – entities that occur within the same article talking about the other entity – entities that occur within the same paragraph – entities that occur within a window of 200 words

Other questions: - how many annotated sentences is enough?

2 Resources

2.1 DBpedia

2.2 DBpedia Spotlight

3 Context Extraction

The context extraction itself consists of processing the properties, objects, types and labels of the DBpedia resource and group them by the subject in order to have different sources of context to model each DBpedia entity. The very first step of such process is to extract all properties , objects and types of DBpedia resources. We consider as potential resources data provided by (1) the Mapping-based properties, as (...), (2) Instance types, as (...) and (3) wikilinks, in which (...). Each context is extracted individually, tokenized and then counted. In the next step, we used the DBpedia Spotlight extraction framework in order to obtain paragraphs from Wikipedias database and theirs corresponding DBpedias resources. From such paragraphs, we striped down words, entities, and so on in order to build an context from the neighborhood. [replace sentence]

4 Context Indexing

5 Related Work

Paper by Ratinov. [?] Paper by Edgar Meij. [?] Paper suggested to me by Johannes Hoffart.

6 Datasets

We will be using the English and Portuguese Wikipedias, and corresponding DBpedias. From those we will use the DBpedia Spotlight extraction framework to obtain paragraphs, and from those paragraphs we will strip down words, grab entities, etc. From the new code we will extract properties, objects, types, etc. from the ontology, to have our "ontology neighborhood" context.

For each kind of context, we will generate one occs.tsv file. T We will generate an index with the Lucene backend (initially) for each of the context files.

7 Experiments

In order to measure the impact of the proposed disambiguation technique, we carried out (...)X evaluation of the context indexing. To this end, we have indexed data from several sources to run distinct automatic evaluation in choosing the correct candidate named entity for a given mention in the text. For the indexing purpose, we used the Dbpedia Spotlight extraction framework, in which we could analyze different scenarios from different context indexing (..). In such process, we were able to compare each individual result against each other, and also run a machine learning algorithm by combining individual context indexing. (...)

We will use the EvaluateParagraphDisambiguation code to run evaluations with the same datasets for each of the context indexes we built. This should give us some initial insight into the tradeoff between the context extraction methods. Report these in a table.

8 Results

We will run the evaluation classes to test which of the methods for context extraction give better disambiguation results. We will also try to combine all of them to see if that performs better than each individual option. Finally, we can see which combinations offer best performance. This can be done by simply concatenating two-three datasets, or more smartly via machine learning. Each dataset provides one score and we use a machine learning algorithm to learn how to combine these scores into one such that disambiguation accuracy is maximized.

9 Conclusion

Potential Conclusions / Impact - Compression: if we find out that we can discard words, we can have the entire index in less space (allowing to load to RAM, perform faster, etc.) - Need for training data: if we find out that using entity names from the neighborhood from an ontology is good enough, then our technique is applicable to several other ontologies without massive corpora like Wikipedia/DBpedia