# Where is the Context for Disambiguation? An Experimental Analysis.

June 19, 2014

## 1 Introduction

In natural language processing, Disambiguation is the process of resolving the ambiguity that arise in entities present in unstructured texts. To be processed, however, we first need to provide a system to solve the annotation of entities in such texts. In this sense, the DBpedia Spotlight [] - an open source project, provides different techniques to recognize and disambiguate natural language mentions in the Web. The Natural Language Processing community has put a considerable effort to solve the disambiguation problem, as many applications now rely on natural language. In fact, current and open problems that could benefit from advances in this area are, for instance, context-sensitive disambiguation, social ads, and many other related problems. From the social media perspective, disambiguation is even more critical, mainly due to the fact that they are more connected and increasingly presents different types of sources.

The DBpedia Spotlight supports many different languages including English and Portuguese, and its disambiguation process involves structured data from the Dbpedia[] and is highly based on Wikipedia. The DBpedia[] project, provides structured data from the extraction of structured information from Wikipedia, and both DBpedia and Wikipedia offers data in many different langue. Because of this variety of languages, it is also possible to adapt the Dbpedia Spotlight to those languages. On the other hand, working with Wikipedia data brings an overhead due to the size os its dataset, as English samples can easily be very large. In fact, if we restrict the disambiguation problem to work only on ontologies that are associated, like Wikipedia datasets, we would be dealing strictly with large amount of data besides being indirectly discarding other ontologies sources.

Bearing this in mind, this paper aims to evaluate others knowledge bases rather than Wikipedia. Though fixed set of target knowledge bases, context modeling paradigm as well as part of speech descriptors we aim to evaluate the impact of these different source of context in the performance evaluation. To this end, we use different structured DBpedia files, that it is of relevance to the disambiguation process, to extract the entities' context, and verbs that are related to each entity, from the Wikipedia paragraphs. The DBpedia files used are Infobox, Mapping-based properties, Instance Types and Wikilinks, which are discussed later in section X. After this extraction, we generate small different datasets to perform the evaluation itself.

By evaluation each extracted dataset as well as each combination of these datasets, we show that, for English data, X and Y have the best performance when performing the M7M evaluation corpus. Indeed, thought the results, which are better described in section Y, we show that a large amount of space can be saved if you are willing to compromise X% of your accuracy. As full disambiguation process, by using Wikipedia resources, uses XX space and presents UU accuracy. Moreover, we also believe that using this methodology, we enable the use of any ontology as source of context, as long as it has properties, types and objects available, to build an entity name annotation and disambiguation model.

## 2 Related Work

There is an extensive literature covered by the Natural Language Processing community on entity annotation and disambiguation in general. Mendes et al [], Bunesco and Pasca [], Cucerzan [], Mihalcea and Csomai (Wikify!) [19] and Witten and Milne (MW) [20], use text from Wikipedia to learn how to annotate. While Mendes takes into consideration all Wikipedia articles, evaluating many different classes (over X, including person, organization, place, albums, etc), Bunesco and Pasca only evaluate articles under the people by occupation category, and Cucerzan's and Wikify!'s conservative spotting only annotate 4.5 and 6 of all tokens in the input text, respectively.

Recent work in entity annotation and disambiguation has emphasized a contextual approach, in which resources are represented as context words, which, in turn, presents a semantic relation with the resource itself. Common to the traditional approaches, which uses Wikipedia as source, is the idea that a word (resource) is explicit represented in a context-based fashion. Thus, instead of analyzing all data in wich a resource appear, we can extract its context, as they believe resources have semantic connections. In this sense, Jabeen et all[] proposes a solution for the disambiguation problem by using context information. As DBpedia, it also uses Wikipedia to extract one single term as the resource context. Indeed, they explored Wikipedia hyperlinks structure and senses for disambiguating the context of a term pair by learning the meaning of a term using the other term as the context. Thus, Jabeen carried out experiments using Wikipedia disambiguation and hyperlink information, in which Wikipedia disambiguation pages are used to extract all listed senses as candidate senses and populate them in the context set of each input term, and Wikipedia hyperlink page to calculate the semantic similarity between candidate contexts.

Annotation and Disambiguation techniques for social media have, in recent years, gained attention from the NLP community to decide the real meaning of small microblogging posts. Due to the limited content, informal nature and creative language usage, recent research relies on the fact that microblogging are highly contextualized, to proposed contextual approaches to solve the annotation and disambiguation problemas. Accordingly, Edgar Meij proposes a solution to the problem of determining the meaning of a microblog post through semantic linking. It approaches the task of linking tweets, which presents a small content, to Wikipedia concepts as a ranking problem. Thus, given a tweet, a ranked list of concepts meant by or contained in it, is calculated. In such a ranked list, the higher rank indicates a higher degree of relevance of the concept

to the tweet. In fact, this approach uses Wikipedia article pages to determine the context of each tweet to facilitate the disambiguation task.

Inspired by these different researches that achieve significant results by using context to enrich the learning data, we propose something that has not yet been experimented in the literacture. In order to find out how important different semantic relations are to a specific resource, we carried out different experiments by using different semantic relations type to find out the influence and significance of each context type in the disambiguation task. **TODO**: ...

# 3 Resources

## 3.1 DBpedia Ontology

The DBpedia [**?**] is one of the most well known knowledge bases in the SeWeb community. It is a project that is interlinking different sources in the Linked Open Data cloud by organizing the knowledge extracted from Wikipedia in a structure of 320 different classes in an hierarchy fashion. Those classes are described by 1,650 different properties and has been growing even more. The English knowledge base holds labels and abstracts for more then 4 million resources classified in a consistent cross-domain ontology with classes such as of persons, places, music albums, films, video games, organizations, species, diseases, among others. Besides the resources classification itself, it also provides links to external web pages, images, Wikipedia categories and geographic coordinates for places. In fact, DBpedia provides a rich pool of resource attributes and relations between the resources. In addiction, the DBpedia sets use a large multi-domain ontology which has been derived from Wikipedia and comprises, among others, datasets such as Mapping-based Types, Mapping-based Properties, Titles, Short and Extended Abstracts, etc. These datasets are what describe each of tis resources, in terms of types, categories, normalized properties, etc. The DBpedia Spotlight, as already stated, uses source from Wikipedia and from DBpedia datasets. The key ideia of the context extraction, however, relies on the fact that we can build different sources of context to model each DBpedia entity, discarding the Wikipedia dependency. Through the DBpedia datasets, we can extract words from different resources such as properties, objects, types and labels and group them by the subject. Accordingly, as one of the main goal of this paper is also to minimize the data used in the learning process, we consider as potential resources data provided by the following sets:

**Infobox** dataset stands as a complete coverage of all Wikipedia properties, which represents all properties from all infoboxes and templates within all articles from Wikipedia. It uses the http://dbpedia.org/property/ namespace to represent extracted information, which directly reflect the name of the Wikipedia infobox property. For the 3.8 version, there are approximately 8000 different property types.

**Mapping-based properties**, in turn, it is a hand-generated mappings of Wikipedia infoboxes/templates. The major difference between Mapping-based and Infobox datasets relies in the fact that the Mapping-based is much cleaner and better structured than Infobox. However, due to this extraction method, Mapping based doesn't cover all infobox types and infobox properties within Wikipedia as Infobox does.

**Instance types**, also known as Mapping-based Types, like the Mapping-based properties dataset, is a hand-generated mappings of Wikipedia infoboxes/templates, which extracts the types instead of properties. It is a DBpedia Ontology RDF type statements, that are extracted from infoboxes.

**Wikilinks** dataset contains all intern links between DBpedia resources, which is the result of the extraction of all internal links between Wikipedia articles.

### 3.2 Wikipedia

Since DBpedia has its roots on Wikipedia, previous work [?] has used Wikipedia paragraphs to model DBpedia Resources for disambiguation tasks. In the DBpedia Spotlight perspective, for each DBpedia resource, it is extracted the list of paragraphs containing links to the corresponding Wikipedia page for this specific resource. The paragraphs contained within the list is then tokenized, cleaned by removing stopwords, and aggregated with the term frequencies over the entire corpus for each resource. Thus, to learn an annotation model in the DBpedia Spotlight Wikipedia is still needed. Therefore, as an evaluation experiment, we have extended previous work by inspecting more closely different sources of paragraphs, in order to determine wether the full Wikipedia paragraphs are necessary. As we can identify many potential sources of context to model each DBpedia entity, this paper aims to evaluate wether we can break a paragraph and use only its properties, objects, etc. To this end, from the list of potential sources os context, we are evaluating the text from DBpedia ontology.

- title (the title of the article associated with concept c),
- sentence (the first sentence of the article),
- definition (the first paragraph of the article),
- article (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: i) all properties, ii) all objects, iii) etc.

## 4 Approach

### 4.1 Context Knowledge

Previous work[][], as well as related work[][] normaly learn a annotate model from full datasets such as Wikipedia, FreeBase[] or any other complete knowledge base. These KB are composed of full texts, which contains different source of knowledge such as title, abstract, sentence, the article itself, among others different contentx, which may vary depending on the KB itself. During the DBpedia Spotlight's indexing process, we run a full statistic analysis on the artigle itself, considering all different source of knowledge present in the input dataset. Thus, once all the entity names are identified, a massive statistc is run on each Wikipedia paragrah, in order to calculate different metrics to reach

the best model without dismiss any type of information. Context Knowlede, on the other hand, contains a set of entities with the semantic relations among different words, that is, we can extract different part of text, different words that are somehow correlated, from many potencial sources.

As already stated, Dbpedia provides different dataset which cover different aspect of the Wikipedia resources. The Input Types set, for instance, presents all types thar are related to a specific resource, as well as the Mapping based properties offers the mapping of different relations among different resources. Bearing this in mind, we can extract such relations and build a context of all DBpedia labels containing properties, objects and types, for example.

### 4.2 Building the Context Model

In order to evaluate these different datasets, our approach follows three steps. The first step is to process the RDF representation of DBpedia to extract all properties, objects, types and labels of DBpedia resources. We highlight that It is not all DBpedia datasets that contain all these different categories. For instance, the Instance Types dataset only holds types of an specific resource, as well as the Labels only holds the subjects. In the second step, we group all extracted words by the subject extracted from the label dataset. After the grouping stage, each extracted context is tokenized and then counted. At this step, we have all data needed to learn the model, as we are exchanging the Wikipedia pre-processing, which has been better explained in the section T, by this context data. After the context data processing, at the third step, we learn individual annotation models from each dataset created separately to finally, at the final step, evaluate each of these datasets and combine them together, in order to trace the model with the best performance. At the extraction stage, as a complement to the object context extraction, we highlight that properties extraction was also covered for both Mapping-Based Properties and Infobox, as such datasets cover information at both levels - object and property.

### 4.3 Evaluation Corpus

Milne CSAW CoNLL

## 5 Experiments

Our aim was to evaluate if our annotate system can provide a faster and flexible indexing, by using considerable smaller set of ontologies data, at the same time it present a more accurate performance in comparison with the default DBpedia Spotlight indexing method. Faster, in the sense that we are now able to proccess less data, and use less memory to run the indexing task itself. Flexible, since we can provide didfferent ways to index different ontologies rather than Wikipedia, as long as structured ontologies are provided (just like DBpedia ontologies). And finally, having both advantages without compromising the good accuracy that DBpedia Spotlight offers.

The evaluation was carried out on the DBpedia Spotlight framework by using DBpedia ontologies for English language. We also carried out indexing and evaluation on Verbs Extraction from Wikipedia Paraghaps. Thus, we evaluated

indexing and runtime performance, input data size and annotation accuracy. We have used three different evaluate corpus, XX, YY and ZZ.

Accordingly, in order to measure the impact of the proposed disambiguation approach, we carried out distinct evaluations on the different contexts models. The context models extracted from he DBpedia ontologies are Infobox, Mapping based, Page Links and Instance types. And for each one of these datasets, we created models composed of objects, properties and both- objects and properties. As evaluate corpus we have used well known datasets such as Milne, CSAW and CoNLL, so that we can have a better understanding of the performance of all model. To this end, we have indexed data from several context source to run distinct automatic evaluation in choosing the correct candidate named entity for a given mention in the text, which is the annotation task itself . After running the indexed model with each of the evaluate corpus, we compare each individual result against each other, and **TODO**: also run a machine learning algorithm by combining individual context indexing results. (...) **TODO**: We have not done this yet.

Thereby, to evaluate each result, we have processed the contexts models bearing in mind the following questions: (i) Are words (types, objects, properties, etc) from the ontology the most important ones for describing each entity? (ii) Can we safely concentrate on those words for effective disambiguation? and (iii) Are there some sources of paragraphs in Wikipedia that are better than others? Insofar we build the experiment validation in each one of the context sets, we intend to answer all these questions.

# 6 Results

## 6.1 DBpedia Spotlight Model

The results for the baselines and DBpedia Spotlight are presented in Table 1. The performance of the baseline that makes random disambiguation choices confirms the high am- biguity in our dataset (less than 1/4 of the disambiguations were correct at random). Using the prior probability to choose the default sense performs reasonably well, being ac- curate in 55.12

In order to better understand the significance of each context dataset in the disambiguation task, we compare each result against the DBpedia Spotlight model, which is run by processing both Wikipedia and DBpedia resources. Thus, our basis for comparison is the DBpedia Spotlight results that are presented in Table A. The performance of the DBpedia Spotlight model confirms the high accuracy and a high memory usage for preproccesing data, as we show afterwards, by considering all the data processed to learn this model. The amount of data, in termos of space, needed to build the DBpedia Spotlight is also present in Table A. A comparison among space and accuracy of all context models are shown in the next tables.

Table 1: DBpedia Spotlight Model Accuracy/Space

| Ontology Dataset | Milne | CSAW | CoNLL |
|---|---|---|---|
| Accuracy | x | x | x |
| Space | | xxx | |

### 6.2 Ontology-based context

Table 2: Objects Context Accuracy

| Ontology Dataset | Milne | CSAW | CoNLL |
|---|---|---|---|
| Infobox | 0.607 | 0.575 | 0.623 |
| MappingBased | 0.576 | 0.508 | 0.549 |
| Pagelinks | | | 0.311 |
| InstanceTypes | 0.583 | 0.520 | 0.514 |

Table 3: Properties Context Accuracy

| Ontology Dataset | Milne | CSAW | CoNLL |
|---|---|---|---|
| Infobox | 0.617 | 0.562 | 0.567 |
| MappingBased | 0.611 | 0.500 | 0.514 |
| Pagelinks | - | - | - |
| InstanceTypes | - | - | - |

Table 4: Objects and Properties Context Accuracy

| Ontology Dataset | Milne | CSAW | CoNLL |
|---|---|---|---|
| Infobox | 0.590 | 0.582 | 0.633 |
| MappingBased | 0.604 | 0.507 | 0.551 |
| Pagelinks | - | - | - |
| InstanceTypes | - | - | - |

### 6.3 Context Models Space

### 6.4 Paragraph Verbs context

### 6.5 Optimal combinations

### 6.6 Extracting words from the ontology or from paragraphs?

## 7 Advanced / Future

### 7.1 Use full paragraphs or only entity names?

When modeling entity context for disambiguation from paragraphs, one of the following will happen:

1. entity names are the most important part of the text, therefore we can safely discard other words

2. all words provide context that entity names cannot capture

Table 5: Context Models Space

| | Infobox | MappingBased | Pagelinks | InstanceTypes | Paragraph Verbs |
|---|---|---|---|---|---|
| Space | 0 | 0 | 0 | 0 | 0 |

Table 6: POS Accuracy

| Ontology Dataset | Milne | CSAW | CoNLL |
|---|---|---|---|
| left tokens | | | |
| right tokens | | | |
| right and left tokens | | | |

3. Not black and white: there is a trade-off when discarding words. Discarding low frequency words does not hurt performance

4. Not black and white: there is a trade-off when discarding words. Discarding high frequency words does not hurt performance

5. (can we test if words are distributionally independent from entity names)

### 7.2 Where to get words or entity names?

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

Other questions: - how many annotated sentences is enough?

### 7.3 How much linguistic knowledge helps?

#### POS

#### Constituency Parses

## 8 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