Cross-language context extraction for disambiguation

Renan Dembogurski

No Institute Given

Abstract.

1 Introduction

Among the open data communities responsible for the Semantic Web development, one of the most important is DBpedia. As their main site states, DBpedia is a is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web. In fact, the DBpedia project releases an extraction framework used to extract the structured information contained in the infoboxes and to convert it in triples.

In fact, every DBpedia resource can be mapped into types in the onthology (Person, Location, etc.). This association is done through mappings (between onthology and infoboxes) created manually that are incomplete for various reasons. Among the various mappings provided by DBpedia, one in specific is the dataset linking a DBpedia resource to the same resource in other languages, called inter-language links.

With this dataset it is possible to create an information extraction approach that can be later used to enrich a language in various aspects. One of the important aspects that can be improved by this approach is the types coverage in a certain language.

questions/problems: - how to identify/map entities from each language? - Freebase types are enough for this process? - Is there any drawback?

hypotheses: - an inter-language approach provides better accuracy

2 Related Work

As a rich and free resource, Wikipedia contains very large amount of articles written in different languages and various types of link information showing the relations between articles. It has been used as external resource in many natural language processing tasks successfully ([2], [4], [3]). Among the link information in Wikipedia, the inter-language link, which is created by article authors, connects large amount of articles that describe the same term but are written in different languages

Besides of article titles, there still exists large amount of information that could be used for dictionary construction, such as the text inside the linked articles.

//Conferir - It has been demonstrated that these inter-language links are useful resources for bilingual dictionary construction ([3]).

Papers:

- Bilingual Dictionary Extraction from Wikipedia

- Automatic Expansion of DBpedia Exploiting Wikipedia Cross-Language Information

- Cross-Language Information Retrieval with Latent Topic Models Trained on a Comparable Corpus

3 Methodology

At the time of starting the experiments, there are 360 mappings available for the English DBpedia, covering around 1.7M entities, against almost 4M articles in Wikipedia. The remaining pages are automatically mapped to the trivial top-level class owl:Thing. Hereafter, when we speak about coverage, we will always refer to classes different from owl:Thing. (tirei de um artigo, talvez atualizar e usar no nosso?)

3.1 Entity Representation

Our approach exploits the Wikipedia cross-language links to represent each entity. We use an approach similar to [1] where we represent entities using an entity matrix.

(aqui inicia o texto desse autor, podemos nos basear) Formally, we proceed as follows to automatically derive the set of sentities ϵ, also used to build the training set. Let $\mathcal{L}$ be the set of languages available in Wikipedia, we first build a matrix E where the i-th row represents an entity $e_i \in E$ and j-th column refers to the corresponding language $l_j \in \mathcal{L}$. The crosslanguage links are used to automatically align on the same row all Wikipedia articles that describe the same entity. The element $E_{i,j}$ of this matrix is null if a Wikipedia article describing the entity e_i does not exist in l_j. An instance in our machine learning problem is therefore represented as a row vector e_i where each j-th element is a Wikipedia article in language l_j. Table 1 shows a portion of the entity matrix.

(interessante eu acabei modificando o arquivo do interlanguage igual esse autor, podemos justificar a escolha pela matriz dele)

In our case we want to find the first group of types for a resource. We search its main language for types and, if we don't find types for it, we also search addition languages.

3.2 Datasets

We will be using several files to fulfill our task of Cross-Language enrichment. These files are provided by DBpedia and can be described in two groups of files:

en	de	it	. . .
Xolile Yawa	Xolile Yawa	null	. . .
The Locket	null	Il segreto del medaglione	. . .
Barack Obama	Barack Obama	Barack Obama	. . .
null	null	Giorgio Dendi	. . .
. . .	. . .	. . .	. . .

Table 1: A portion of the entity matrix

- The first group has a single file, the interlanguage links triples file. This file is responsible of linking a resource in one language to another resource in another language using the predicate *sameAs*. From now on we will refer to this file as **ILF**. An example of this relation can be seen in Table 2 (colocar tabela).
- The second group is composed of triples files with types for each resource in a certain language. From now on we will reference these files as **instance types files**. The relation between a resource and its types is described by the predicate *rdf:type*. The user determines the number of languages to be used.

(TABELA 2)

The ILF is modified twice to become a file in the matrix representation presented in 1 as a pre-processing step (**ILFC**). Our experiments have shown that this ensures less processing when using our greedy algorithm as will be seen in later sections.

4 Methodology

The methodology adopted for the Cross-Language enrichment is describe in Figure (colocar figura). First we get our two groups of files as input, then we proceed with the filtering, conforming, search and appending stages.

Since we want to complement the types of new resources in the main language instance types file, we look for new resources in the ILFC triples file. For each line in the latter file we will remove entries we already know, in other words, resources that have types already in their language instance types file. This reduces redundancy and prevents duplication of resources. One interesting factor of this filtering process is that if a user selects few languages as complement, the input file can drop 10 times in size.

After the filtering stage, we proceed with the conforming stage. This second stage transforms the ILF from a lines to a columns format file (ILFC), with every column representing a language as shown in 1. This format have a few advantages:

- We do not need to allocate all the instance types files into memory, we can process them by language (column). In the initial lines format we do not know what languages have types for a certain resource, so we need to allocate multiple instance types files into memory in case we need to query them.

- We can easily apply sequential filters to the ILFC file to reduce processing. As we end querying a language instance types file for types we can remove lines/columns from the ILFC file that won be used when querying types in the next language. An example of the sequential filtering can be seen in Figure (colocare figura).

5 Experiments

...

6 Conclusion

Potential Conclusions / Impact
- Is it necessary a “transitive closure” method?
- Drawbacks?
- How complete are the Freebase types?

References

1. Aprosio, A.P., Giuliano, C., Lavelli, A.: Automatic expansion of dbpedia exploiting wikipedia cross-language information. In: Cimiano, P., Corcho, s., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC. Lecture Notes in Computer Science, vol. 7882, pp. 397–411. Springer (2013), `http://dblp.uni-trier.de/db/conf/esws/eswc2013.html#AprosioGL13`
2. Buscaldi, D., Rosso, P.: Mining knowledge from wikipedia for the question answering task. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC). pp. 727–730. Genoa, Italy (2006)
3. Erdmann, M., Nakayama, K., Hara, T., Nishio, S.: An approach for extracting bilingual terminology from wikipedia. In: Proceedings of the 13th international conference on Database systems for advanced applications. pp. 380–392. DASFAA’08, Springer-Verlag, Berlin, Heidelberg (2008), `http://dl.acm.org/citation.cfm?id=1802514.1802552`
4. Mihalcea, R.: Using Wikipedia for Automatic Word Sense Disambiguation. In: North American Chapter of the Association for Computational Linguistics (NAACL 2007) (2007)