VUA-background : When to Use Background Information to Perform Word Sense Disambiguation

Marten Postma
VU University Amsterdam

Ruben Izquierdo
VU University Amsterdam

Piek Vossen
VU University Amsterdam

{m.c.postma, ruben.izquierdobevia, p.t.j.m.vossen}@vu.nl

Abstract

We present in this paper our submission to task 13 of SemEval2015, which makes use of background information and external resources (DBpedia and Wikipedia) to automatically disambiguate texts. Our approach follows two routes for disambiguation: one route is proposed by a state–of–the–art WSD system, and the other one by the predominant sense information extracted in an unsupervised way from an automatically built background corpus. We reached 4th position in terms of F1-score in task number 13 of SemEval2015: "Multilingual All-Words Sense Disambiguation and Entity Linking" (Moro and Navigli, 2015). All the software and code created for this approach are publicly available on GitHub[1].

1 Introduction

Word Sense Disambiguation is still an unsolved problem in Natural Language Processing. Many different approaches have been proposed throughout the years to tackle this task from different perspectives. In addition, competitions have been organized to compare the performance of these approaches. Our hypothesis is that, in general, the context is not being modelled properly by the systems, which usually consider very narrow contexts and do not pay any attention to the background information or information that is not explicitly included in the text. We conducted an in-depth error analysis of previous all-words tasks (Senseval–2 : English all words (Palmer et al., 2001), Senseval–3 : English all words (Snyder and Palmer, 2004), Semeval–2007 : all words task 17 (Pradhan et al., 2007), Semeval–2010 : all words task 17 (Agirre et al., 2010), Semeval–2013 : all words task 12 (Navigli et al., 2013)) in order to gain better insight as to why some approaches perform better than others, to detect problems not properly addressed and to try to overcome them. [2]

We observed that most systems tend to rely on **local features** (words surrounding the words in question) to perform word sense disambiguation. Besides this, there is a very acute trend by all WSD systems to assign in most cases the most frequent sense, regardless the domain under consideration, as can be seen in Figure 1:

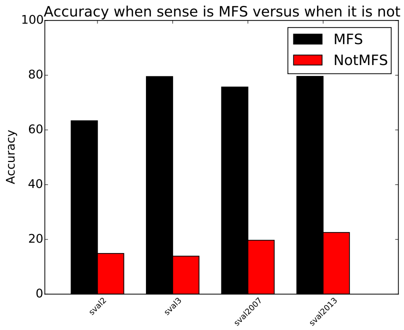

Figure 1: The average accuracy of all systems per competition is shown.

Figure 1 shows the average accuracy of all the systems per competition. We clearly observe the trend that systems perform well when the sense is the most frequent sense, but not in other cases. Furthermore, when the sense is not the most frequent one, the systems still propose the most frequent sense. For instance in Senseval–2, out of 799 tokens for which the correct sense is not the most frequent one, systems still wrongly assign the most frequent sense in 84% of the cases.

Based on these observations, we designed a system that creates background corpora starting from a set of seed documents, from now on SD (preferably from a specific and unique domain). From this

[1] https://github.com/cltl/vua-wsd-sem2015

[2] The error analysis can be found here: https://github.com/cltl/WSD_error_analysis.

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 345–349,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

corpus, we use the entities automatically detected to access DBpedia and create the first background corpus, which will be called Entity Article (EA) corpus. By applying different techniques, we expand this EA corpus with more domain related documents, which results in the Entity Expanded (EE) corpus. Once the whole background corpus (EA+EE) has been created, we use this information to automatically derive the specific predominant sense of each word in our target domain (the domain of the starting documents and also the domain of the background corpus).

The rationale behind this approach starts with the observation that the predominant sense of a lemma is very dominant in a document. Hence, by focusing on when to use or not to use this predominant sense, a high performance seems plausible. In addition, we observed that local features are not always enough to determine the correct sense of a lemma and we should only rely on these features when they are necessary.

The structure of this paper is as follows. We introduce our approach in section 2. followed by the results in section 3. Finally we discuss and conclude our results in section 4.

2 Our Approach

Figure 2 shows the overall architecture of our system, that will be explained more in detail in this section.

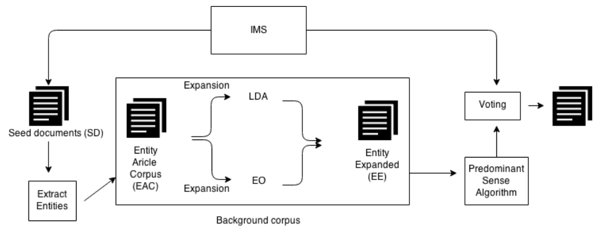

Figure 2: Overall architecture

Seed documents: We focused on the WSD part of the task. The input for our approach is a collection of seed documents, which represent the target domain that is used for calculating the predominant domain information. These documents can either be the task test documents (**online approach**) or a different set of documents that we could compile in advance if the target domain is known (**offline approach**). We first converted these documents to the NAF format (Fokkens et al., 2014).[3]. We then applied a POS–tagger to get the lemmas and part-of-speech labels for all the tokens. As explained before, two different approaches were followed: online and offline. We experimented with both approaches and finally the online approach was selected for our participation due to the mixed-domain nature of the test documents. The documents follow two different and parallel routes of analysis: one route which favors the domain predominant sense by using the **background knowledge** and one route which favors the most frequent sense (in a general domain) by using one of the state–of–the–art WSD systems that performs very well in such domains. Finally, a voting heuristic of the two routes is applied to assign the final senses.

2.1 Route 1: Background knowledge

Extract entities from collection of documents We started with one corpus of documents (the test documents in the online approach or a pre–compiled set in the offline version): the seed documents (SD). Then we applied the statistical implementation of DBPedia Spotlight (Daiber et al., 2013) in order to obtain entities and their corresponding links to DBPedia[4]. With this we compile the EA corpus, which contains all the Wikipedia texts associated to the DBpedia links extracted[5]. We experimented with some filtering techniques on the list of DBPedia links in order to keep just domain specific ones, such as considering only those DBPedia links tagged with an ontological concept which is a leaf of the ontology tree. Nevertheless, we found a better performance when using all the DBpedia links without any filtering.

[3] `http://www.newsreader-project.eu/files/2013/01/techreport.pdf`

[4] We developed our own module that calls automatically the DBPedia Spotlight end–point and allows to work with NAF files: `http://github.com/rubenIzquierdo/dbpedia_ner`

[5] We also created our own modules to query DBPpedia (`http://github.com/rubenIzquierdo/dbpediaEnquirerPy` based on SPARQL) and WikiPedia: `https://github.com/rubenIzquierdo/wikipediaEnquirerPy`

Expansion The EA corpus generated in the previous section represents the domain of our test data (online/offline), but probably suffers from a low coverage, especially for our idea of applying a predominant sense algorithm which relies on the availability of a large domain corpus. In order to expand this EA corpus, we developed two strategies to generate the EE corpus: a) Latent Dirichlet Allocation–based (LDA), targeting a high recall and low precision/quality, and b) Entity overlapping (EO), aiming a high quality and medium/low recall.

The **LDA technique** first obtains a topic model using LDA on the EA corpus[6]. This is our domain model for comparison. Moreover, we obtain the DBPedia ontology classes for all the documents in the EA corpus (one example could be *HumanGene*). For each of these labels, DBPedia is queried to get all the entities belonging to that label (following our example, all the entities that are *HumanGene*)[7]. The Wikipedia text for every of these entities is gathered and compared against the LDA model obtained previously. Only those reaching a certain similarity are selected to be part of the EE corpus. The whole process is highly time consuming and the result in terms of quality is not as good as expected, probably related to the fact that the number of documents retrieved is very high, the domains are very diverse and in many cases different to our reference domain.

The **EO expansion** follows a different approach. On the one hand, all the DBpedia entities in the EA corpus are extracted, which makes up our set of domain entities (DE). On the other hand, each of the Wikipedia pages that can be reached from these DE is processed to extract all the possible wiki–links. All these wiki–links are possible candidates for the EE corpus. To select the final set of candidates, the similarity is obtained by measuring the overlap between the wiki–links of the candidate with our initial domain set DE. Only those surpassing a certain overlapping threshold are selected.

Predominant sense algorithm Our background corpus is considered the union of the EA and the EE corpus, which usually is a large collection of NLP-processed documents. For each lemma in these documents, we extract all the sentences containing this lemma. If there are at least 100 sentences, we feed the sentences for this specific lemma into the predominant sense algorithm. The predominant sense algorithm we use is based on topic modeling (Lau et al., 2012; Lau et al., 2014). The algorithm first tries to induce senses using a Hierarchical Dirichlet Process and then tries to determine the sense ranking of all senses of a lemma according to the documents. The output of this step is a list of sense confidences for each lemma for which we had enough training data. [8]

2.2 Route 2: it–makes–sense WSD system

Our idea is to start from the output of a state-of-the-art WSD system, and combine it with the predominant sense information automatically gathered with our approach, in order to obtain an overall WSD approach specific to our target domain. We selected the it–makes–sense system (Zhong and Ng, 2010) that has proved to be one of the best performing WSD systems in general domains. Similarly, we have created our own wrapper around the it–makes–sense system that allows the use of NAF format as input/output for this tool[9]. Following our purpose, we did not only select the most likely sense in each case according to the WSD engine, but we stored all the possible senses for each lemma along with the probability returned by it–makes–sense.

2.3 Voting

For each token in the test data, we first check if we have predominant sense output for this lemma. In addition, we check if the sense ranking is skewed, which we determine by checking if the two senses with the highest confidence have a combined confidence of higher than 85%. If this is the case, we calculate the average of the sense rankings of the predominant sense output and the it-makes-sense sys-

[6] We have used the Python library GenSim for this purpose `http://radimrehurek.com/gensim/`

[7] This process can be quite time consuming (there are a total of 15 entries in DBpedia for *HumanGene*, but there are 1.65 million entries for *Person*)

[8] we created a wrapper around the GitHub repositories that were created to run the predominant sense algorithm (`https://github.com/jhlau/hdp-wsi`, `https://github.com/jhlau/predom\_sense`). This github can be found at `https://github.com/MartenPostma/predominantsense`

[9] This wrapper module can be found at `http://github.com/rubenIzquierdo/it_makes_sense_WSD`

tem and choose the sense with the highest confidence. If we do not have predominant sense output, we assign the sense with the highest confidence according to the it-makes-sense system. Finally, we did not provide answers to all instances in the test set due to the fact we used an older version of WordNet, which did not contain all the gold senses. These lemmas mainly consisted of computer related senses.

3 Results

The results can be found in Table 1:

All_domains

Measure	all	n	v
Precision	67.5 (2)	64.7	56.6
Recall	51.4 (5)	42.9	53.9
F1	58.4 (4)	51.6	55.2

Social_issues_domain

Measure	all	n	v
F1	61.1 (2)	54.8 (7)	70.6 (1)

Math_Computer_domain

Measure	all	n	v
F1	47.7 (5)	30.5 (13)	49.7 (7)

Biomedical_domain

Measure	all	n	v
F1	66.4 (4)	62.7 (9)	53.8 (2)

Table 1: Results of VUA-background are shown for the domains: 'All', 'Social_issues', 'Math_Computer, and 'Biomedical'. The results per domain are presented for all part of speeches, as well as for nouns and verbs. The numbers in parentheses are competition ranks.

As can be seen in Table 1, our system finished 4nd in terms of F1-score, 2nd in terms of precision, and 5th in terms of recall. In particular. our system performed well on the biomedical domain and the Social_Issues domain, and mainly for verbs. In addition, running the evaluation using only the predominant sense output led to an improvement in the precision for nouns (69.1% versus 64.7%) and verbs (61.6% versus 56.6%), but also a drop in recall for both nouns (20.1% versus 42.9%) and verbs (17.7% versus 53.9%).

4 Discussion and Conclusion

A number of reasons have attributed to the fact that our system performed relatively well in terms of precision, but not so well in terms of recall.

Firstly, our system, and in particular our offline approach, is built around the notion of one dominant theme or topic. The domain of this evaluation was announced to be the biomedical domain, but the test documents ended up belonging to several domains, which has hurt the performance of our algorithm. We believe that adapting our system to work with multiple domains is the next step in improving the algorithm.

In addition, our system was built around WordNet 1.7.1. This means that we did not provide answers to all instances, which has had an impact on the recall.

Finally, we claim that size is an issue in obtaining good results. Especially our online approach could have benefited from more data.

We presented a WSD framework that exploits both information available in a document or a set of documents, and background information from different external resources. We believe the results achieved in this evaluation task are promising, despite the problems and issues mentioned in the previous paragraphs. Our approach is especially suited to deal with one single domain, or with a domain that is known in advance. We will continue working on the adaptation of the whole framework to a multi–domain scenario. Furthermore, all software developed is publicly available on different GitHub repositories. Our system can be found at `https://github.com/cltl/vua-wsd-sem2015`. Scripts are included, which will run the whole process step by step starting from the official test documents and apply: linguistic processors (tokenizer, lemmatizer), entity detection, linking to DBpedia, call to it–makes–sense system, creation of the background corpus and expansion, creation of the predominant sense information and final voting heuristic.

Acknowledgments

The research for this paper was supported by the Netherlands Organisation for Scientific Research (NWO) via the Spinoza-prize Vossen projects (SPI 30-673, 2014-2019).

References

Eneko Agirre, Oier López de Lacalle, Christiane Fellbaum, Shu-Kai Hsieh, Maurizio Tesconi, Monica Monachini, Piek Vossen, and Roxanne Segers. 2010. Semeval-2010 task 17: All-words Word Sense Disambiguation on a Specific Domain. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 75–80, Uppsala, Sweden, July.

Joachim Daiber, Jakob Max, Chris Hokamp, and Pablo N. Mendes. 2013. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 121–124, New York, NY, USA.

Antske Fokkens, Aitor Soroa, Zuhaitz Beloki, Niels Ockeloen, German Rigau, Willem Robert van Hage, and Piek Vossen. 2014. NAF and GAF: Linking linguistic annotations. In *Proceedings 10th Joint ISO-ACL SIGSEM Workshop on Interoperable Semantic Annotation*, pages 9–16, Reykjavik, Iceland.

Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word Sense Induction for Novel Sense Detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 591–601, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2014. Learning Word Sense Distributions, Detecting Unattested Senses and Identifying Novel Senses Using Topic Models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 259–270, Baltimore, Maryland, USA, June 23-25.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In *Proc. of SemEval-2015.*

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA, June.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 21–24, Toulouse, France, July.

Sameer S. Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 Task-17: English Lexical Sample, SRL and All Words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic, June.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain, July.

Zhi Zhong and Hwee Tou Ng. 2010. H.t.: It makes sense: A wide-coverage Word Sense Disambiguation system for free text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL*, pages 78–83.