Iryna Gurevych

Darmstadt, Germany

UKP-WSI: UKP LAB SEMEVAL-2013 TASK 11 SYSTEM DESCRIPTION

Introduction. The task «Evaluating Word Sense Induction and Word Sense Disambiguation in an End-User Application» of SemEval-2013 (Navigli and Vanella, 2013) aims at an extrinsic evaluation scheme for WSI to overcome the difficulties inherent to WSI evaluation. The task requires building a WSI system and combining it with a WSD step to assign the induced sentences to example instances.

Word sense disambiguation (WSD) is the task of determining the correct meaning for an ambiguous word from its context. WSD algorithms usually choose one sense out of a given set of possible senses for each word. A resource that enumerates possible senses for each word is called a sense inventory. Manually created inventories come usually in form of lexical semantic resources, such asWord-Net or more specifically created inventories such as OntoNotes (Hovy et al., 2006).

Word sense induction (WSI) on the other hand aims to create such an inventory from a corpus in an unsupervised manner. For each word that should be disambiguated, a WSI algorithm creates a set of context clusters that will be used to define and describe the senses.

We build our system upon the open-source DKPro framework 1 and a corresponding WSD component.

Input for the task comes as two files. One contains the search queries, also referred as topics. Sense induction will be performed for each of those topics. The second file contains 6400 entries from the result pages of a search engine. Each entry consists of the title, a snippet and the URL of the corresponding web page.

Related Work. One of the early approaches to WSI (Sch¨utze, 1998) maps words into a vector space and represents word contexts as vector-sums and use cosine vector similarity, clustering is performed by expectation maximization (EM) clustering. Dorow and Widdows (2003) use the BNC to build a co-occurrence graph for nouns, based on a co-occurrence frequency threshold. They perform Markov clustering on this graph. Pantel and Lin (2002) proposes a clustering approach called clustering by committee (CBC). This algorithm first selects the words with the highest similarity based on mutual information and then builds groups of highly connected words called committees. It then iteratively assigns the remaining words to one of the committee clusters by comparing them to the averaged the committee feature vectors. This exploits the assumption that two or more words together disambiguate each other, Bordag (2006) extends on this idea by using word triples to form non-ambiguous seed-clusters. Many approaches use a variety of graph clustering algorithms for WSI: Others (Klapaftis and Manandhar, 2010a; Klapaftis and Manandhar, 2010b) use hierarchical agglomerative clustering on hierarchical random graphs created from word co-occurrences. Di Marco and Navigli (2012) use word sense induction for web search result clustering. They introduce a maximum spanning tree algorithm that operates on co-occurrence graphs built from large corpora, such as WaCky (Baroni et al., 2009). The system by Pedersen (2010) employs clustering firstand second-order co-occurences as well as singular value decomposition on the co-occurrence matrix, which is clustered using repeated bisections. Jurgens (2011) employ a graph-based community detection algorithm on a co-occurrence graph. Distributional approaches for WSI include LSA Van de Cruys and Apidianaki (2011) or LDA (Brody and Lapata, 2009).

Our Approach. Our system consists of two independent parts. The first is a batch process that creates database containing co-occurrence statistics derived from a background corpus. The second is the actual WSI and WSD pipeline doing the result clustering. Both parts include identical preprocessing steps for segmentation and lemmatization.

The pipeline (Figure 1) first performsWord Sense Induction, resulting in an induced sense inventory. A WSD algorithm then uses this inventory to disambiguate all instances of the search query within a document. Finally a result writer will produce the cluster mappings used by the evaluation system. It uses a majority vote on all instances of the target word within the snippet.

The sense induction algorithm is based on graph clustering on a co-occurrence graph, similar to the approach by Di Marco and Navigli (2012). Our approach differs from previous work in the way we perform a greedy search for additional context and how it combines WSI with an advanced WSD step using lexical expansions. Moreover, we consider our generic UIMA-based WSD and WSI system as (Tabl.1) a useful basis for experimentation and evaluation of WSI systems.

Table 1

Size of co-occurrence databases

	# words	# co-occurrences
Wikipedia	3,011,397	96,979,920
WaCky	8,687,711	441,005,478

Preprocessing

The topics and snippets are read by a custom collection reader. If the web-page can be downloaded at the URL that corresponds to the result, it is cleaned by an HTML parser and the plain text is appended to the snippet. As further steps we segment and lemmatize the input. We apply the same preprocessingto snippets, queries and the corpora.

Co-occurrence Extraction

We calculate the log-likelihood ratio (LLR) (Dunning, 1993) and point-wise mutual information (PMI) (Church and Hanks, 1990) of a word pair co-occurring at sentence level using a modified version of the collocation statistics implemented in Apache Mahout. Even when sorting the co-occurrences by PMI, we employ a minimum support cut-off based on the LLR, all pairs with a log-likelihood ratio < 1 are discarded. Table 1 gives an overview about the obtained co-occurrence pairs.

Clustering Algorithm

The algorithm is a two-step approach that first creates an initial clustering of a graph *G = (V;E)* and then improves this clustering in a second step. The initial step (Algorithm 1) starts by retrieving the top n = 150 most similar terms for the target word by querying the co-occurrence database we created in section 3.2. These represent vertices in a graph. We then construct a minimum spanning tree *(mst)* by inserting edges fvi; vjg from the co-occurrence database. The weight *w(fvi; vjg*) of each edge is set to the inverse of the used similarity measure dist (LLR or PMI) between those terms. The minimum spanning tree then is cut into subtrees be iteratively

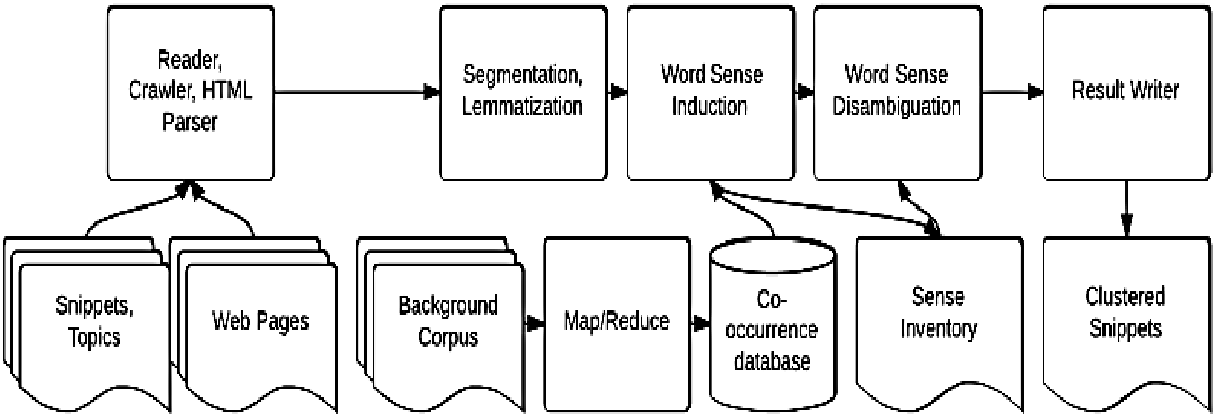

Figure 1: WSI and WSD Pipeline removing the edge with the highest edge betweenness (betweeness) (Freeman, 1977) until the size of the largest component of G falls below a threshold $S_{initial}$.

Algorithm 1 initialClusters

```
V (G0) top n most similar words to target word
w(vi; vj) dist(termi; termj)
G mst(G0)
V (G) V (G) n vtarget
while max(jC(G)j) > Sintitial do
    E(G) E(G) n argmaxe(betweeness(e))
end while
```

The resulting partitioning of the graph is the starting point for the second phase of the algorithm, which we call *expand/join* step (Algorithm 2). During this step, the algorithm looks iteratively at all clusters of size s smaller than $S_{max} = 9$ (determined empirically), starting with the largest ones. From each of these clusters, it constructs a query to the co-occurrence database, retrieving all terms that significantly co-occur together with all terms in the respective cluster and with the target word. This list of terms is then compared to all clusters C_{large} with $|C| > s$. If the normalized intersection between one of those C_{large} is above a threshold $t = 0{:}3$ (determined empirically), we assume that the C_{small} represents the same sense as the Clarge and merge those clusters. If this is not the case for any of the larger clusters, we assume that C_{small} represents a sense of its own extend the cluster by adding edges between vertices representing the expansion terms and C_{small}.

Algorithm 2 expandJoin

```
Require: G is a minimum spanning forest
for $s = S_{max}$ → 1 do
    for all $C_{small}(G)$; $|C_{small}| = s$ do
        expansions ← querys($v_1$; ::; $v_i$)
        for all components $C_{large} \in G$; $|C_{large}| > s$ do
            if $|C_{large} \cap expansions| / |C_{large}| > t$ then
                $C_{large} \leftarrow C_{large} \cup C_{small}$
            else
                $C_{small} \leftarrow C_{small} \cup expansions$
            end if
        end for
    end for
end for
```

Word Sense Disambiguation

We use the DKPro WSD framework, which implements various WSD algorithms, with the same system configuration as reported by Miller et al. (2012). It uses a variant of the Simplified Lesk Algorithm (Kilgarriff et al., 2000). This algorithm measures the overlap between a words context and the textual descriptions of senses within a machine readable dictionary, such as WordNet. The senses that have been induced in the previous step are provided to the framework as a sense inventory. Instead of using sense descriptions, we now compute the overlap between the sense clusters and the context of the target word. The WSD system expands both the word

context and the sense clusters with synonyms from a distributional thesaurus (DT), similar to Lin (1998). The DT has been created from 10M dependency parsed sentences of English newswire for word similarity. Besides knowledge-based WSD, the DT also has been successfully used for improving the performance of semantic text similarity (Bar et al., 2012). The WSD component disambiguates each instance of the search query within the snippet and web page individually.

Table 2

Results for the submitted runs

Run	F1	ARI	RI	JI	# clusters	avg cl. size
wacky-llr	0.5826	0.0253	0.5002	0.3394	3.6400	32.3434
wp-llr	0.5864	0.0377	0.5109	0.3177	4.1700	21.8702
wp-pmi	0.6048	0.0364	0.5050	0.2932	5.8600	30.3098

Results. The clustering was evaluated using four different metrics as described by Di Marco and Navigli (2012). The Rand index and its chance-adjusted variant ARI are common cluster evaluation metrics. The adjusted rand index gives special weight to less frequent senses. The Jaccard index (JI) disregards the cases where two results are assigned to different clusters in the gold standard, therefore it is less sensitive to the granularity of the clustering. The F_1-Measure gives more attention to the individual clusters and how they cover the topics in the gold standard.

We submitted several runs for different configurations of the co-occurrence database (Table 2). Between runs, we did not modify the configuration of the sense induction or disambiguation step. The first run used collocations extracted from WaCky scored by LLR metric (wacky-llr), and two others used Wikipedia as background corpus. One of the Wikipedia based runs used PMI as association metric (wp-pmi), the other one used LLR (wp-llr). The run on the larger ukWac corpus scored best with regard to the Jaccard measure, but worst in the adjusted Rand index measure. We attribute low scores for ARI to the fact that our system did not induce certain less frequent senses, resulting in small average number of clusters. The coarse grained clusters however, have been assigned quite well by our WSD system, as shown by relatively high Jaccard Index. For the Wikipedia-based runs, the clustering based on PMI produced more clusters and therefore scored higher on the F_1 measure than the LLR-based run. Due to an error in the implementation our submissions contained only a single sense for 20% of the senses. However, we repeated our experiments with a correctly configured system which produced only slightly different results.

Conclusion

We presented our word sense induction and disambiguation pipeline for search result clustering. Our contribution is a sense induction algorithm that incrementally retrieves more context from a co-occurrence database and the integration of WSI and WSD into a UIMA-based pipeline for easy experimentation.

The system scored best with regard to Jaccard similarity of clusters, while performing low especially with the Adjusted rand index. We assume that our sense granularity was too low for this task and failed to create clusters for rare senses. This could be improved by making the merge phase of the induction algorithm less eager. Furthermore, increasing the size of the background corpus, e.g. by combining the both corpora that have been used could increase the size of the context clusters especially for rare senses, which should further improve the performance in these cases. We attribute the good results with regard to the F1 and Jaccard measures also to our state-of-the-art word sense disambiguation step and the use of the distributional thesaurus.

Acknowledgements

We thank Tristan Miller for helping us with the DKPro WSD framework and Chris Biemann for providing the distributional thesaurus. This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806.

References:

1. Daniel Bar, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures. In Proceedings of First Joint Conference on Lexical and Computational Semantics (*SEM), pages 435—440.
2. Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Language Resources and Evaluation, 43(3):209—226, February.
3. Stefan Bordag. 2006. Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation. In Proceedingsof the 11th Conference of the European Chapter of the Association for Computational Linguistics, pages 137—144, Trento, Italy.
4. Samuel Brody and Mirella Lapata. 2009. Bayesian Word Sense Induction. Computational Linguistics, (April):103—111.
5. Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. Computational Linguistics, 16(1):22—29, March.
6. Antonio Di Marco and Roberto Navigli. 2012. Clustering and Diversifying Web Search Results with Graph-BasedWord Sense Induction. Computational Linguistics, pages 1—46, November.
7. Beate Dorow and Dominic Widdows. 2003. Discovering corpus-specific word senses. In Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - EACL ’03, volume 2, page 79, Morristown, NJ, USA, April. Association for Computational Linguistics. Ted Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics, 19(1):61 — 74.
8. Linton C Freeman. 1977. A set of measures of centrality based on betweenness. Sociometry, 40(1):35—41.
9. Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. pages 57—60, June.
10. David Jurgens. 2011. Word Sense Induction by Community Detection. In HLT ’11: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics Human Language Technologies, pages 24—28, Portland, Oregon.
11. Adam Kilgarriff, Brighton England, and Joseph Rosenzweig. 2000. English Senseval: Report and Results. In Proceedings of the 2nd International Conference on Language Resources and Evaluation, Athens, Greece.
12. Ioannis P. Klapaftis and Suresh Manandhar. 2010a. Taxonomy learning using word sense induction. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, number June, pages 82—90. Association for Computational Linguistics.
13. Ioannis P. Klapaftis and Suresh Manandhar. 2010b. Word sense induction & disambiguation using hierarchical random graphs. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 745—755. Association for Computational Linguistics, October.
14. Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In Proceedings of the 36th annual meeting on Association for Computational Linguistics, volume 2, pages 768—774, Morristown, NJ, USA, August. Association for Computational Linguistics.
15. Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 2012. Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation. In Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012).
16. Roberto Navigli and Daniele Vanella. 2013. SemEval-2013 Task 11: Evaluating Word Sense Induction & Disambiguation within An End-User Application. In Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantcis (*SEM 2013), Atlanta, USA.

17. Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’02, page 613, New York, New York, USA, July. ACM Press.

18. Ted Pedersen. 2010. Duluth-WSI: SenseClusters applied to the sense induction task of SemEval-2. In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 363—366, Stroudsburg, PA, USA, July. Association for Computational Linguistics.

19. Hinrich Schutze. 1998. Automatic word sense discrimination. Computational Linguistics, 24(1):97—123, March.

20. Tim Van de Cruys and Marianna Apidianaki. 2011. Latent semantic word sense induction and disambiguation. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 1476—1485, Portland, Oregon, June. Association for Computational Linguistics.

In this paper, we describe the UKP Lab system participating in the Semeval-2013 task «Word Sense Induction and Disambiguation within an End-User Application». Our approach uses preprocessing, co-occurrence extraction, graph clustering, and a state-of-theart word sense disambiguation system. We developed a configurable pipeline which can be used to integrate and evaluate other components for the various steps of the complex task