Distributional Lesk: Effective Knowledge-Based Word Sense Disambiguation

Dieke Oele
CLCG
Rijksuniversiteit Groningen
The Netherlands
d.oele@rug.nl

Gertjan van Noord
CLCG
Rijksuniversiteit Groningen
The Netherlands
g.j.m.van.noord@rug.nl

Abstract

We propose a simple, yet effective, Word Sense Disambiguation method that uses a combination of a lexical knowledge-base and embeddings. Similar to the classic Lesk algorithm, it exploits the idea that overlap between the context of a word and the definition of its senses provides information on its meaning. Instead of counting the number of words that overlap, we use embeddings to compute the similarity between the gloss of a sense and the context. Evaluation on both Dutch and English datasets shows that our method outperforms other Lesk methods and improves upon a state-of-the-art knowledge-based system. Additional experiments confirm the effect of the use of glosses and indicate that our approach works well in different domains.

1 Introduction

The quest of automatically finding the correct meaning of a word in context, also known as Word Sense Disambiguation (WSD), is an important topic in natural language processing. Although the best performing WSD systems are those based on supervised learning methods (Snyder and Palmer, 2004; Pradhan et al., 2007; Navigli and Lapata, 2007; Navigli, 2009; Zhong and Ng, 2010), a large amount of manually annotated data is required for training. Furthermore, even if such a supervised system obtains good results in a certain domain, it is not readily portable to other domains (Escudero et al., 2000).

As an alternative to supervised systems, knowledge-based systems do not require manually tagged data and have proven to be applicable to new domains (Agirre et al., 2009). They only require two types of information: a set of dictionary entries with definitions for each possible word meaning, and the context in which the word occurs. An example of such a system is the Lesk algorithm (Lesk, 1986) that exploits the idea that the overlap between the definition of a word and the definitions of the words in its context can provide information about its meaning.

In this paper, we propose a knowledge-based WSD method that is loosely based on the Lesk algorithm exploiting both the context of the words and the definitions (hereafter referred to as glosses) of the senses. Instead of counting the number of words that overlap, we use word- and sense embeddings to compute the similarity between the gloss of a sense and the context of the word. The strong point of our method is that it only requires large unlabeled corpora and a sense inventory such as WordNet, and therefore does not rely on annotated data. Also, it is readily applicable to other languages if a sense inventory is available.

2 Related work

In the past few years, much progress has been made on learning word embeddings from unlabeled data that represent the meanings of words as contextual feature vectors. A major advantage of these word embeddings is that they exhibit certain algebraic relations and can, therefore, be used for meaningful

semantic operations such as computing word similarity (Turney, 2006), and capturing lexical relationships (Mikolov et al., 2013).

A disadvantage of word embeddings is that they assign a single embedding to each word, thus ignoring the possibility that words may have more than one meaning. This problem can be addressed by associating each word with a series of sense-specific embeddings. For this, several methods have been proposed in recent work. For example, in Reisinger and Mooney (2010) and Huang et al. (2012), a fixed number of senses is learned for each word that has multiple meanings by first clustering the contexts of each token, and subsequently relabeling each word token with the clustered sense before learning embeddings.

Although previously mentioned sense embedding methods have demonstrated good performance, they use automatically induced senses. They are, therefore, not readily applicable to NLP applications and research experiments that rely on WordNet-based senses, such as machine translation and information retrieval and extraction systems (see Morato et al. (2004) for examples of such systems). Recently, features based on sense-specific embeddings learned using a combination of large corpora and a sense inventory have been shown to achieve state-of-the-art results for supervised WSD (Rothe and Schütze, 2015; Jauhar et al., 2015; Taghipour and Ng, 2015).

Our system makes use of a combination of sense embeddings, context embeddings, and gloss embeddings. Somewhat similar approaches have been proposed by Chen et al. (2014) and Pelevina et al. (2016). The main difference to our approach is that they automatically induce sense embeddings and find the best sense by comparing them to context embeddings, while we add gloss embeddings for better performance. Inkpen and Hirst (2003) apply gloss- and context vectors to the disambiguation of near-synonyms in dictionary entries. Also Basile et al. (2014) use a distributional approach, however, it requires a sense-tagged corpus while our system does not rely on any tagged data.

3 Method

Our WSD algorithm takes sentences as input and outputs a preferred sense for each polysemous word. Given a sentence $w_1 \dots w_i$ of i words, we retrieve a set of word senses from the sense inventory for each word w. Then, for each sense s of each word w, we consider the similarity of its lexeme (the combination of a word and one of its senses (Rothe and Schütze, 2015) with the context and the similarity of the gloss with the context.

For each potential sense s of word w, the cosine similarity is computed between its gloss vector G_s and its context vector C_w and between the context vector C_w and the lexeme vector $L_{s,w}$. The score of a given word w and sense s is thus defined as follows:

$$\text{Score}(s, w) = cos(G_s, C_w) + cos(L_{s,w}, C_w) \quad (1)$$

The sense with the highest score is chosen. When no gloss is found for a given sense, only the second part of the equation is used.

Prior to disambiguation itself, we sort the words by the number of senses is has, in order that the word with the fewest senses will be considered first. The idea behind this is that words that have fewer senses are easier to disambiguate (Chen et al., 2014). As the algorithm relies on the words in the context which may themselves be ambiguous, if words in the context have been disambiguated already, this information can be used for the ambiguous words that follow. We, therefore, use the resulting sense of each word for the disambiguation of the following words starting with the "easiest" words.

Our method requires lexeme embeddings $L_{s,w}$ for each sense s. For this, we use AutoExtend (Rothe and Schütze, 2015) to create additional embeddings for senses from WordNet on the basis of word embeddings. AutoExtend is an auto-encoder that relies on the relations present in WordNet to learn embeddings for senses and lexemes. To create these embeddings, a neural network containing lexemes and sense layers is built, while the WordNet relations are used to create links between each layer. The advantage of their method is that it is flexible: it can take any set of word embeddings and any lexical

database as input and produces embeddings of senses and lexemes, without requiring any extra training data.

Ultimately, for each word w we need a vector for the context C_w, and for each sense s of word w we need a gloss vector G_s. The context vector C_w is defined as the mean of all the content word representations in the sentence: if a word in the context has already been disambiguated, we use the corresponding sense embedding; otherwise, we use the word embedding. For each sense s, we take its gloss as provided in WordNet. In line with Banerjee and Pedersen (2002), we expand this gloss with the glosses of related meanings, excluding antonyms. Similar to the creation of the context vectors, the gloss vector G_s is created by averaging the word embeddings of all the content words in the gloss.

4 Experiments

The performance of our algorithm was tested on both Dutch and English sentences in an all-words setup. Our sense inventory for Dutch is Cornetto (Vossen et al., 2012) while, for English, we use WordNet 1.7.1 (Fellbaum, 1998). In Cornetto, 51.0% of the senses have glosses associated with them and in the Princeton WordNet, almost all of them do. The DutchSemCor corpus (Vossen et al., 2013) is used for Dutch evaluation and, for English, we use SemCor (Fellbaum, 1998). For both languages, a random subset of 5000 manually annotated sentences from each corpus was created. Additionally, we test on the Senseval-2 (SE-2) and Senseval-3 (SE-3) all-words datasets (Snyder and Palmer, 2004; Palmer et al., 2001)[1].

We build 300-dimensional word embeddings on the Dutch Sonar corpus (Oostdijk et al., 2013) using word2vec CBOW (Mikolov et al., 2013), and create sense- and lexeme embeddings with AutoExtend. For English, we use the embeddings from Rothe and Schütze (2015)[2]. They lie within the same vector space as the pre-trained word embeddings by Mikolov et al. (2013)[3], trained on a part of the Google News dataset, which contains about 100 billion words. This model (similar to the Dutch model) contains 300-dimensional vectors for 3 million words and phrases.

We evaluate our method by comparing it with a random baseline and Simplified Lesk with expanded glosses (SE-Lesk) (Kilgarriff and Rosenzweig, 2000; Banerjee and Pedersen, 2002). Additionally, we compare our system to a state-of-the-art knowledge-based WSD system, UKB (Agirre and Soroa, 2009), that, similar to our method, does not require any manually tagged data. UKB can be used for graph-based WSD using a pre-existing knowledge base. It applies random walks, e.g. Personalized PageRank, on the Knowledge Base graph to rank the vertices according to the context. We use UKBs Personalized PageRank method word-by-word with WordNet 1.7 and eXtended WordNet for English, as this setup yielded the best results in Agirre and Soroa (2009). For Dutch, we use the Cornetto database as input graph.

We do not compare our system to the initial results of AutoExtend (Rothe and Schütze, 2015) as they tested it in a supervised setup using sense embeddings as features. However, as is customary in WSD evaluation, we do compare our system to the most frequent WordNet sense baseline, which is notoriously difficult to beat due to the highly skewed distribution of word senses (Agirre and Edmonds, 2007). As this baseline relies on manually annotated data, which our system aims to avoid, we consider this baseline to be semi-supervised and therefore an upper bound.

For Dutch, the manually annotated part of DutchSemCor is balanced *per sense* which means that an equal number of examples for each sense is annotated. It is therefore not a reliable source for computing the most frequent sense. Alternatively, similar to Vossen et al. (2013), we derive sense frequencies by using the automatically annotated counts in DutchSemCor[4], assuming that the automatic annotation

[1]For SenseEval-2, we used conversions from WordNet 1.7 to 1.7.1 from http://web.eecs.umich.edu/ mihalcea/downloads.html

[2]http://www.cis.lmu.de/ sascha/AutoExtend/

[3]https://code.google.com/p/word2vec/

[4]In DutchSemCor senses are annotated with an SVM, trained on the manually annotated part of the corpus, see Vossen et al. (2013) for more details.

sufficiently reflects the true distribution for this purpose. The most frequent sense baseline for Dutch is, therefore, lower as compared to the English one, where the most frequent sense of a word is fully based on manual annotation.

5 Results

The results of the evaluation of our method (Lesk++) for both Dutch and English can be found in Table 1. Accuracy is calculated by dividing the number of words that were disambiguated correctly, as compared to the sense tagged corpus, by the total amount of polysemous words. Results are in bold when statistically significant over the baselines at $p < 0.05$.

	Dutch	English		
	DSC	SC	SE-2	SE-3
ES-Lesk	31.3%	45.2%	47.1%	43.4%
UKB	35.7%	41.2%	51.4%	47.4%
Lesk++	**42.1%**	53.5%	52.1%	49.3%
Random	27.1%	33.6%	35.8%	30.1%
MFS	37.0%	**69.9%**	**59.7%**	**59.5%**

Table 1: Results on DutchSemCor (DSC), SemCor (SC) Senseval-2 (SE-2) and Senseval3 (SE-3)

For both Dutch and English, our method performs significantly better than SE-Lesk and the random baseline for all tasks. Also, our system performs better than UKB on both SemCor and DutchSemCor. On DutchSemCor, it outperforms the most frequent baseline.

5.1 Effects of sorting, lexemes, and glosses

The main idea behind our method is a simple combination of two cosine similarity scores. In a second experiment, we evaluate the effects of both of these scores by using them separately. Additionally, we examine the use of sorting the words by its number of senses before disambiguation.

We compare our final results with a system where similarity is only computed between the context and gloss vector and with a system that only computes the cosine distance between the context and the lexeme (only the first and the second part of Equation 1 respectively). Both systems are tested without and with (+S) sorting. The results of this third experiment on the sense tagged corpora for Dutch (DSC) and English (SC) can be found in Table 2.

	Lesk++	Lex	+S	Gloss	+S
DSC	42.1%	38.3%	38.6%	41.5%	41.6%
SC	53.5%	42.7%	47.0%	52.8%	52.6%

Table 2: Effects of lexemes, glosses and sorting. The second and the fourth column show results of a system that only uses the lexeme (Lex) or gloss vectors (Gloss) respectively. In the third and last column sorting (+S) is added.

For Dutch, the results indicate that sorting the words by its number of senses by itself is not very effective compared to the system that does not use this module. The use of glosses, on the other hand, seems to be very effective while the combination of both measures yields the best results. The effect of the gloss vectors is even stronger for English, which can be explained by the fact that the English WordNet has a higher gloss coverage. Also, for English, although both sorting and glosses are effective, the combination performs better.

5.2 Comparison of Different Domains

To examine the robustness of our system in different domains, we evaluate it on different parts of DutchSemCor. We randomly took 5000 manually annotated sentences from each of the four largest subsets of the corpus. The results of this experiment for the all-words task can be found in Table 3. On every subsec-

	dl	st	wp	np
SE-Lesk	27.7%	30.4%	28.8%	29.4%
UKB	30.5%	32.1%	37.3%	33.8%
Lesk++	**36.8%**	**36.8%**	**45.6%**	40.3%
Random	24.2%	23.5%	28.2%	25.7%
MFS	30.6%	33.3%	35.8%	**42.9%**

Table 3: Results for the Dutch all-words task on a random subset of each of the four largest datasets from DutchSemCor: discussion lists (dl), subtitles (st), Wikipedia (wp) and newspapers (ns).

tion of the DSC dataset, our method outperforms SE-Lesk, the random baseline and UKB. Furthermore, our method outperforms the most frequent baseline on three of them. The newspapers subsection forms an exception, probably because it belongs to a more general domain (Agirre et al., 2014).

6 Discussion

The difference in results for Dutch and English can possibly be explained by the coverage of the datasets. The Cornetto coverage is about 60%, compared to Princeton Wordnet, with an average polysemy of 1.07 for nouns, 1.56 for verbs and 1.05 for adjectives while, for English it is 1.24 for nouns, 2.17 for verbs and 1.40 for adjectives. Also, not all Dutch senses have corresponding glosses while most of the English ones do. As our method relies greatly on gloss vectors, this could affect its performance.

Our WSD approach combines a lexical knowledge base with word- and sense embeddings. The results of our experiments show that the use of embeddings can help improve other Lesk methods (Kilgarriff and Rosenzweig, 2000; Banerjee and Pedersen, 2002). An obvious next step would be to see whether other extensions that do not require manually tagged data are compatible as well. For example, Vasilescu et al. (2004) shows improvements by pre-selecting context words using the WordNet hierarchy. Also, the method of Miller et al. (2012) could be used to first expand the glosses and/or the context before using our adaptation of the Lesk system.

7 Conclusion

In this paper we propose an extension to the Lesk algorithm which uses sense, gloss and context embeddings to compute the similarity of word senses to the context in which the words occur. Although our approach is a straightforward extension to the Lesk algorithm, it achieves better performance compared to Lesk and a random baseline and outperforms, or yields similar performance to, a state-of-the-art knowledge-based system. For Dutch, it outperforms all other systems including the most frequent sense on three out of four subsets. A second experiment confirms the effects of gloss vectors while the results of a final experiment indicate that our method works well in different domains. The main advantage of our method is its simplicity which makes it fast and easy to apply to other languages. It furthermore only requires unlabeled text and the definitions of senses, and does not rely on any manually annotated data, which makes our system an attractive alternative for supervised WSD.

References

Agirre, E., O. L. De Lacalle, and A. Soroa (2009). Knowledge-based WSD on specific domains: Performing better than generic supervised WSD. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, pp. 1501–1506.

Agirre, E. and P. Edmonds (2007). *Word Sense Disambiguation: Algorithms and Applications* (1st ed.). Springer Publishing Company, Incorporated.

Agirre, E., d. O. L. Lacalle, and A. Soroa (2014). Random walks for knowledge-based word sense disambiguation. *Computational Linguistics 40*(1), 57–84.

Agirre, E. and A. Soroa (2009). Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pp. 33–41.

Banerjee, S. and T. Pedersen (2002). An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 136–145.

Basile, P., A. Caputo, and G. Semeraro (2014). An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of the 25th International Conference on Computational Linguistics*, pp. 1591–1600.

Chen, X., Z. Liu, and M. Sun (2014). A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1025–1035.

Escudero, G., L. Màrquez, and G. Rigau (2000). An empirical study of the domain dependence of supervised word sense disambiguation systems. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 172–180.

Fellbaum, C. (Ed.) (1998, May). *WordNet An Electronic Lexical Database*. Cambridge, MA ; London: The MIT Press.

Huang, E. H., R. Socher, C. D. Manning, and A. Y. Ng (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 873–882.

Inkpen, D. Z. and G. Hirst (2003). Automatic sense disambiguation of the near-synonyms in a dictionary entry. In *Computational Linguistics and Intelligent Text Processing, 4th International Conference*, pp. 258–267.

Jauhar, S. K., C. Dyer, and E. H. Hovy (2015). Ontologically grounded multi-sense representation learning for semantic vector space models. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 683–693.

Kilgarriff, A. and J. Rosenzweig (2000, May). English senseval: report and results. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, Athens, Greece. European Language Resources Association (ELRA).

Lesk, M. (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, New York, NY, USA, pp. 24–26. ACM.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. *CoRR*.

Mikolov, T., W. Yih, and G. Zweig (2013). Linguistic regularities in continuous space word representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings*, pp. 746–751.

Miller, T., C. Biemann, T. Zesch, and I. Gurevych (2012). Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In *Proceedings of COLING 2012*, pp. 1781–1796.

Morato, J., M. N. Marzal, J. Llorns, and J. Moreiro (2004). Wordnet applications. In *Proceeding of the Second Global Wordnet Conference*.

Navigli, R. (2009, February). Word sense disambiguation: A survey. *ACM Computing Surveys 41*(2), 10:1–10:69.

Navigli, R. and M. Lapata (2007). Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 1683–1688.

Oostdijk, N., M. Reynaert, V. Hoste, and I. Schuurman (2013). *The Construction of a 500-Million-Word Reference Corpus of Contemporary Written Dutch*, pp. 219–247. Berlin, Heidelberg: Springer Berlin Heidelberg.

Palmer, M., C. Fellbaum, S. Cotton, L. Delfs, and H. T. Dang (2001). English tasks: All-words and verb lexical sample. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pp. 21–24.

Pelevina, M., N. Arefiev, C. Biemann, and A. Panchenko (2016). Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pp. 174–183.

Pradhan, S. S., E. Loper, D. Dligach, and M. Palmer (2007). SemEval-2007 task 17: English lexical sample, SRL and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 87–92.

Reisinger, J. and R. J. Mooney (2010). Multi-prototype vector-space models of word meaning. In *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 109–117.

Rothe, S. and H. Schütze (2015). AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 1793–1803.

Snyder, B. and M. Palmer (2004, July). The english all-words task. In R. Mihalcea and P. Edmonds (Eds.), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pp. 41–43.

Taghipour, K. and H. T. Ng (2015, May–June). Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 314–323.

Turney, P. D. (2006). Similarity of semantic relations. *Computational Linguistics, Volume 32, Number 3, September 2006*.

Vasilescu, F., P. Langlais, and G. Lapalme (2004). Evaluating variants of the Lesk approach for disambiguating words. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*.

Vossen, P., A. Görög, R. Izquierdo, and A. van den Bosch (2012, may). Dutchsemcor: Targeting the ideal sense-tagged corpus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, pp. 584–589.

Vossen, P., R. Izquierdo, and A. Görög (2013). Dutchsemcor: in quest of the ideal sense-tagged corpus. In G. Angelova, K. Bontcheva, and R. Mitkov (Eds.), *Recent Advances in Natural Language Processing*, pp. 710–718.

Vossen, P., I. Maks, R. Segers, H. d. v. Vliet, M.-F. Moens, K. Hofmann, E. T. K. Sang, and M. de Rijke (2013). *Cornetto: A Combinatorial Lexical Semantic Database for Dutch*, pp. 165–184. Berlin, Heidelberg: Springer Berlin Heidelberg.

Zhong, Z. and H. T. Ng (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, pp. 78–83.