Semantic Similarity Functions in Word Sense Disambiguation

Łukasz Kobyliński and Mateusz Kopeć

Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland,
lkobylinski@ipipan.waw.pl, m.kopec@ipipan.waw.pl

Abstract. This paper presents a method of improving the results of automatic Word Sense Disambiguation by generalizing nouns appearing in a disambiguated context to concepts. A corpus-based semantic similarity function is used for that purpose, by substituting appearances of particular nouns with a set of the most closely related similar words. We show that this approach may be applied to both supervised and unsupervised WSD methods and in both cases leads to an improvement in disambiguation accuracy. We evaluate the proposed approach by conducting a series of lexical sample WSD experiments on both domain-restricted dataset and a general, balanced Polish-language text corpus.

1 Introduction

Word Sense Disambiguation (WSD) is now a well known task of computational linguistics, for which many automated methods have already been proposed. It is a problem that consists of assigning the meaning to a given instance of a polysemous word, based on the context, in which it has been used.

In this article we propose a method of improving existing approaches to word sense disambiguation by using available linguistic knowledge resources to generalize information included in contexts of the disambiguated words. We argue that using either a corpus- or relation-based Semantic Similarity Function (SSF, discussed in Section 3) to find lexemes closely related to the words appearing in disambiguated contexts may significantly increase disambiguation accuracy. By using an SSF we include important semantic information in the purely statistical process of selecting the correct sense for a particular word. This benefits both the unsupervised, knowledge-based approaches to WSD (as described in Section 4) by increasing the chances of matching a particular context with a sense definition and supervised methods, in which case the contexts extended with semantically related words translate to richer training material for machine learning methods, as we describe in Section 5. In Section 6 we provide results of experiments validating the proposed approach by comparing original WSD methods and methods extended with an SSF.

2 Previous work

As reported by the organizers of public evaluations of WSD methods (e.g. [1]), supervised learning approaches currently achieve the highest accuracy in the task of WSD. This class of approaches requires that a training corpus is available, annotated with information about the sense in which each or some of the words appear in the text. Unsupervised methods, which use external knowledge sources, such as WordNet [2] or Wikipedia and unsupervised learning approaches, can be used in situations where very little or no training data in the form of annotated corpus is available. For example in [3] a graph-based approach has been presented, where WordNet has been used as a lexical knowledge base containing hierarchical information about relationships between ambiguous words and other elements of the language. In the context of Polish language, an approach to WSD, which involved 106 polysemous target words and a large corpus of more than 30 000 instances has been described in [4]. A WSD method based on mining class association rules has been presented in [5].

An idea related to the one presented in this contribution, which concerns the expansion of training data with WordNet parents, has been proposed in [6]. Lesk method has been modified to use WordNet relations in [7], while in [8] a distributional similarity has been used to expand the sense definitions.

3 Semantic Similarity Functions

Semantic similarity function (SSF) is defined as a mapping from pairs of words (or lemmas) into real numbers: $W \times W \to \mathbb{R}$. The value of this function for a given pair of semantically strongly related words should be greater than the value for another pair of words, between which the relation is weaker. For example, $SSF(book, page)$ should be greater than $SSF(book, pen)$, because although both pairs show some kind of relatedness, the former is arguably stronger. If we take the type of the linguistic resource (used to extract the similarity of two words) into consideration, semantic similarity functions can be broadly divided into two types: distribution- (or corpus-) based and relation-based (see for example [9] and [10], respectively).

Corpus-based similarity functions rely on the following idea: the more often two words occur in similar context (and are used in similar way), the more semantically similar they are. Based on their frequency or more sophisticated statistical features of the contexts, in which they occur or co-occur, a real number representing the similarity may be calculated. The second type of semantic similarity functions relies on the existence of structural linguistic resources such as WordNet. They traverse the semantic network (or any taxonomy) between words to calculate the value of the function. A large number of WordNet SSFs have been developed, but their requirement is the availability of such a large structured language resource, which makes these methods less applicable for languages other than English.

For the Polish language, there were only a few attempts to create similarity functions. In this paper we use a corpus-based Rank Weight Function

(RWF, [11]), to find nouns most closely related to the ones appearing in disambiguated contexts. Further in this paper, we are going to refer to this function simply as semantic similarity function (or SSF).

4 Extending the Knowledge-Based Approach to WSD

Although most often the supervised WSD techniques achieve the highest accuracy, the need to investigate unsupervised methods still exists, because of one main reason: lack of sufficient amount of training data. As every disambiguated word needs its own set of training examples, the *all-words* WSD task is most often best conducted by the knowledge-based methods, instead of the machine learning algorithms. In this paper we evaluate an extension of the simplified Lesk algorithm, with and without the help of the semantic similarity function.

The proposed extension builds on the idea of comparing the coincidence of sense definition with the context (*Coincidence(i, c)*, where c – context, i – i-th sense definition) and choosing the sense for which the value of the function is the highest. In the Extended Lesk approach the coincidence function is calculated in a more complex way than intersecting the sets of words (as in original Lesk algorithm). Individual steps of the algorithm are presented below.

1. Create two empty maps: W_i (for sense i) and W_c (for context c). They will store pairs: *(lemma, weight)*, *lemma* being the base form of a word, *weight* being a real number.
2. Choose the size of context window (i.e. number of tokens before and after polysemous word to take into account). Tested sizes are: 1, 2, 5, 10, 30, 50.
3. Insert the base forms of words from the dictionary definition of sense i and from the context window into W_i and W_c respectively. Each entry should have a weight equal to its number of occurrences in definition/context.
4. Multiply each weight of each lemma by its Inverse Document Frequency (IDF). In the case of the words from context, frequency is based on the corpus, treating the corpus text as a document. In the case of words from sense definition, IDF is calculated treating all possible sense definitions of currently disambiguated word as the set of documents.
5. Remove from both maps entries with outlying weights, high or low (defined as having higher/lower value than a chosen percentage of the highest one). Tested threshold can be 0% or 1% in case of low outliers, 99% and 100% in case of high ones.
6. Normalize definition of sense i, by dividing each weight in W_i by the number of words in this definition. It prevents bias to longer definitions.
7. Normalize context by dividing the weights from W_c by values dependent on the textual distance of word from the disambiguated one. Three options can be tested – no normalization, division by the distance, division by the squared distance.
8. Extend both maps with related words extracted using Semantic Similarity Function by choosing the words with the highest score with a chosen similarity threshold. For example, if there is a word w in W_c, we extend W_c with

20 words most similar to w, regarding SSF. If a threshold is set, we only add these words from the top 20, which acquire result higher than the threshold. We have tested 4 threshold values: 0.0, 0.1, 0.2, 0.3 and a version, in which we do not extend maps with related words.

9. Compare the maps using product measure or the Jaccard coefficient.

In this way we acquire a single real number, representing the coincidence between the sense i and the context c. The sense with the maximum value is chosen as the correct one. The *Coincidence* function can have a large number of variants, depending on the choice of parameters[1]. Each of these versions produces an individual disambiguation method.

5 Extending the Supervised Learning Approach to WSD

In the supervised learning approach to WSD we use an annotated text corpus to train state-of-the-art machine learning methods and then measure their classification accuracy using the ten-fold cross-validation approach. The fundamental problem we face using machine learning methods for WSD is the selection of an adequate feature representation method, which allows us to express the knowledge about an ambiguous word and its context in the form of a feature vector. We thus transform the WSD task into a classification problem and represent the textual data in the form of fixed-length number vectors.

We have chosen the following representation, implemented as feature generators in the WSD Development Environment [12]. Thematic Feature Generator (TFG): represents the existence of a word in a window around the disambiguated lexeme with window size: 5–25, lemmatization: on/off, generation of related words using a semantic similarity function: on/off and SSF threshold value: 0.1, 0.2, 0.3, or 0.4. Structural Feature Generator 1 (SFG1): existence of a word on a particular position in a small window relative to the disambiguated lexeme with window size: 1–5 and lemmatization: on/off. Structural Feature Generator 2 (SFG2): existence of a part-of-speech on a particular position in a small window relative to the disambiguated lexeme with window size: 1–5 and tagset: full or simplified. Keyword Feature Generator (KFG): grammatical form of the disambiguated lexeme with tagset: full or simplified.

Examples of feature vectors created by the generators described above are presented on Figure 1. We follow the common approach of representing the context of a particular polysemous word with a variant of the bag-of-words representation. The TFG generator captures the information about the existence of a particular word or lemma in the context, while the SFG1 generator analyzes a smaller window around the disambiguated lexeme and adds information about the position of the word in context relative to the lexeme. We also use the

[1] During the development of the Extended Lesk method more extensions were tested than are presented in this paper. We describe only the parameter values, which were found to be the most successful.

TFG	cena	złotówka	moralność	kilogram
	0	1	0	1

SFG1	siebie-1	surowiec+1	praca+1
	1	1	0

SFG2	praet-2	subst-1	adj-1	subst+1
	1	0	0	1

KFG	subst	sg	pl	dat	acc
	1	0	1	0	1

Fig. 1. Examples of feature vectors used with supervised learning WSD methods.

SFG2 generator, which is analogous to SFG1, but takes parts-of-speech appearing in context into account and KFG, which notes the grammatical form of the disambiguated lexeme.

We have used the SSF to extend the TFG generator and to include the information about the words most similar to the words appearing in context in the final feature vector. This allows us to train a more general classifier, which is not closely tied to a particular word, but rather to a general concept. For example, in case of the word "kilogram", the used SSF returns such closely related words as: "kg" (similarity rating: 0.299), "kilo" (0.287), "tona" ("ton", 0.241) and "gram" (0.206). All such words, having the similarity rating above a selected threshold are appended as additional attributes to the feature vector.

The bag-of-words approach to text representation produces a very large number of attributes, which is impractical in the subsequent classifier learning phase. In order to reduce the size of the feature vectors we employ an attribute selection method (still using the training data set), which chooses between 50 to 400 most important attributes, by calculating their information gain with respect to the class.

6 Experimental Results

6.1 Evaluation Corpora

Evaluation of the proposed improvement to automatic WSD has been performed on two corpora, each having its own dictionary of polysemous words (nouns, verbs and adjectives). The larger of the corpora comes from the National Corpus of Polish (NCP) project, described in [13]. It contains 1 215 513 tokens, including 34 114 polysemous ones, in 3 889 texts. It is a balanced corpus, spanning multiple types of textual sources and thematic domains. To verify the performance of the proposed approach on a domain-restricted collection of documents, we have used the Econo corpus, presented in [5]. It consists of 370 182 tokens with 22 520 polysemous ones in 1 861 texts from the domain of economy. Each of the corpora has been manually sense-annotated by qualified linguists (each example was annotated independently by two annotators and an additional third annotator in case of a disagreement) to serve as a verification data set and training material for supervised learning methods. In case of the NCP corpus the sense inventory contained 106 polysemous words and 2.85 sense definitions per word on average. The smaller Econo corpus was annotated using a dictionary of 52 polysemous words and containing 3.62 sense definitions per word on average.

6.2 Unsupervised methods

Based on a combination of parameters of *Coincidence* function (see Table 1), two sets of methods were defined. The first set (EL) was formed by creating all combinations of possible parameter values without the usage of Semantic Similarity Function, which resulted in total number of 288 methods. The second set (EL-SS) consisted of 1152 methods, as the usage of SSF with different parameter values was included. The only difference between these two sets lies in the use of SSF and a chosen similarity threshold, while all the other parameters remain unchanged.

We have used the following experimental framework for the available data. Each corpus was split into two parts of similar size (on the level of texts). The first part was designated as the development part, while the other as the evaluation part. All methods from both sets were tested on the development part. Based on the results from that test, the best methods from EL and EL-SS sets for each lexeme were chosen, as well as single methods, which performed best when used on all of the lexemes. Final results are calculated on the evaluation part using best methods from the previous step.

Table 1. Unsupervised methods achieving the highest accuracy on all lexemes from the inventory (as measured on the development part).

Parameter	Possible values	NCP	Econo
Context size	1,2,5,10,30,50	50	10
Low threshold	0.01,0.00	0.01	0.01
High threshold	1.00,0.99	1.00	0.99
Definition normalization	yes, no	yes	no
Context normalization	none, linear, square	square	none
Comparison measure	product, jaccard	jaccard	product
SSF	yes, no	yes	yes
SSF threshold	0.0, 0.1, 0.2, 0.3	0.1	0.2

6.3 Supervised methods

In the case of supervised methods, a best performing method has been chosen for each of the lexemes in the sense inventory. The selection of the most accurate classification method has been done by searching through the space of feature representation methods, their parameters (as described in Section 5), machine learning algorithms (e.g. NaiveBayes, C4.5, RandomForest) and the number of selected attributes. All the experiments have been performed using the ten-fold cross-validation approach. The accuracy figures given in the following section are calculated by choosing the best supervised method for each of the 52 (in case of Econo corpus) or 106 (in case of the NCP corpus) disambiguated words.

6.4 Results

To assess the improvement in disambiguation accuracy gained from the proposed approach, we have experimented individually with each of the words found in the Econo and NCP sense inventories. The improvement varies greatly between the lexemes, ranging from no improvement to an increase of more than 60 percentage points. Disambiguation accuracy of the polysemous lexemes from the Econo corpus has been presented on Figure 2. Context generalization using the SSF proved to increase the accuracy of disambiguation for 33 polysemous words in case of the unsupervised approach and 19 in case of the supervised methods.

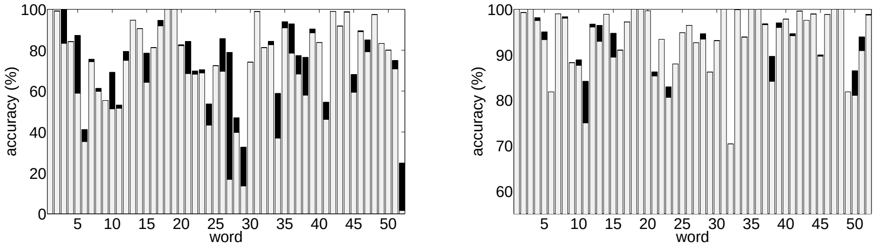

Fig. 2. Disambiguation accuracy of the Econo corpus using the extended Lesk method (left) and supervised learning approach (right) and the increase in accuracy gained using the semantic similarity function (black bars).

The improvement is also considerable in case of the NCP corpus, as presented in Table 2. The parameters of single unsupervised methods that performed best on all the lexemes have been presented in Table 1. In case of the supervised methods the use of a SSF has influenced the disambiguation accuracy of lexemes, which appear less often in the dataset and for that reason the improvement of overall results for this group of approaches is less significant than for the Extended Lesk method.

Table 2. Overall results of supervised and unsupervised methods on Econo and NCP corpora.

Econo corpus

Method	Method set	Accuracy
Single best	EL	60.04%
	EL-SS	61.70%
Best per lexeme	EL	77.27%
	EL-SS	80.92%
	BestSupervised	97.29%
	BestSupervised-SS	97.52%

NCP corpus

Method	Method set	Accuracy
Single best	EL	62.46%
	EL-SS	65.64%
Best per lexeme	EL	75.09%
	EL-SS	80.03%
	BestSupervised	91.47%
	BestSupervised-SS	91.94%

7 Conclusions and Future Work

In this contribution we have described our experiments concerning Word Sense Disambiguation performed on two Polish language corpora: a general, balanced NCP corpus and domain-restricted Econo corpus. We have presented the results of a knowledge-based and supervised learning approaches to WSD in these corpora and proposed and improvement applicable to any method relying on context to perform the disambiguation. In future we would like to explore the possibilities of combining the proposed SSF extension of WSD methods with the knowledge available in WordNet-like resources, to fine-tune the generalization of words in disambiguated contexts.

References

1. Pradhan, S., Loper, E., Dligach, D., Palmer, M.: Semeval-2007 task-17: English lexical sample srl and all words. In: Proceedings of SemEval-2007. (2007)
2. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)
3. Agirre, E., Soroa, A.: Personalizing PageRank for word sense disambiguation. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, ACL (2009) 33–41
4. Kopeć, M., Młodzki, R., Przepiórkowski, A.: Word Sense Disambiguation in the National Corpus of Polish. Prace Filologiczne **vol. LX** (2012) Forthcoming.
5. Kobyliński, Ł.: Mining class association rules for word sense disambiguation. In: Proceedings of the International Joint Conference on Security and Intelligent Information Systems. Volume 7053 of Lecture Notes in Computer Science., Springer-Verlag (2011) 307–318
6. Kohomban, U.S., Lee, W.S.: Learning semantic classes for word sense disambiguation. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL (2005) 34–41
7. Banerjee, S., Pedersen, T.: An adapted Lesk algorithm for word sense disambiguation using WordNet. In: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing. CICLing '02, London, UK, UK, Springer-Verlag (2002) 136–145
8. Iida, R., McCarthy, D., Koeling, R.: Gloss-based semantic similarity metrics for predominant sense acquisition. In: Proceedings of the Third International Joint Conference on Natural Language Processing. (2008) 561–568
9. Lin, D.: Automatic retrieval and clustering of similar words. In: COLING-ACL. (1998) 768–774
10. Budanitsky, A., Hirst, G.: Evaluating WordNet-based measures of lexical semantic relatedness. Computational Linguistics **32**(1) (March 2006) 13–47
11. Piasecki, M., Szpakowicz, S., Broda, B.: Automatic selection of heterogeneous syntactic features in semantic similarity of polish nouns. In: Proceedings of the 10th International Conference on Text, Speech and Dialogue. Volume 4629 of Lecture Notes in Artificial Intelligence., Springer-Verlag (2007) 99–106
12. Młodzki, R., Przepiórkowski, A.: The WSD development environment. In: Proceedings of the 4th Language and Technology Conference. (2009)
13. Przepiórkowski, A., Bańko, M., Górski, R.L., Lewandowska-Tomaszczyk, B., eds.: Narodowy Korpus Języka Polskiego. Wydawnictwo Naukowe PWN, Warsaw, forthcoming.