Word Sense Disambiguation using Statistical Models and WordNet

Antonio Molina, Ferran Pla, Encarna Segarra, Lidia Moreno

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València (Spain)
{amolina,fpla,esegarra,lmoreno}@dsic.upv.es

Abstract

One of the main problems in Natural Language Processing is lexical ambiguity, words often have multiple lexical functionalities (i.e. they can have various parts-of-speech) or have several semantic meanings. Nowadays, the semantic ambiguity problem, most known as Word Sense Disambiguation, is still an open problem in this area. The accuracy of the different approaches for semantic disambiguation is much lower than the accuracy of the systems which solve other kinds of ambiguity, such as part-of-speech tagging. Corpus-based approaches have been widely used in nearly all natural language processing tasks. In this work, we propose a Word Sense Disambiguation system which is based on Hidden Markov Models and the use of *WordNet*. Some experimental results of our system on the *SemCor* corpus are provided.

1. Introduction

Over the last few years, inductive or corpus-based approaches have been widely used in nearly all the Natural Language Processing (NLP) tasks. The availability of linguistic resources such as corpora or dictionaries has made the application and development of these learning techniques possible. These methods have been successfully applied to solve different disambiguation problems, such as part-of-speech (POS) tagging, shallow parsing or chunking, prepositional phrase attachment, etc., using different formalisms: Hidden Markov Models (HMM), transformation-based learning, memory-based learning, decision trees, maximum entropy, etc.

A POS tagger attempts to assign the corresponding POS or morpho-syntactical tag to each word in a sentence, taking into account the context in which this word appears. However, Word Sense Disambiguation (WSD) consists of selecting the semantic sense of a word from all the possible senses given by a dictionary, as well as taking into account the context in which this word appears. Although a WSD problem can be carried out as a POS tagging task, in practice, the former is more difficult and complex than the latter. First, there is no consensus on the concept of sense, and consequently, different semantic tag sets can be defined. In addition, the size of this set is very large compared to the POS tag set and the few available semantic corpora do not have enough annotated data. Second, the modeling of contextual dependencies is more complicated because a large context is generally needed and sometimes the dependencies among different sentences must be known in order to determine the correct sense of a word (or a set of words). Also, the lack of common evaluation criteria makes it very hard to compare different approaches. In this respect, the knowledge base *WordNet* (Miller et al., 1990) and the *SemCor*[1] corpus (Miller et al., 1994) are the most frequently used resources. SENSEVAL[2] competition can be viewed as the most important reference point for WSD.

There has been a wide range of approaches to the WSD problem (a detailed study can be found in (Ide and Véronis, 1998) and (Resnik and Yarowsky, 2000)). In general, you can categorize them into knowledge-based and corpus-based approaches. Under the knowledge-based approach the disambiguation process is carried out using information from an explicit lexicon or knowledge base .

The lexicon may be a machine-readable dictionary, such as the *Longman Dictionary of Contemporary English*, thesaurus, such as *Rodget's Thesaurus*, or large-scale hand-crafted knowledge bases, such as *WordNet* (Lesk, 1986; Yarowsky, 1992; Voorhees, 1993; Resnik, 1995; Agirre and Rigau, 1996; Stevenson and Wilks, 2001).

Under the corpus-based approach, the disambiguation process is carried out using information which is estimated from data, rather than taking it directly from an explicit knowledge base. In general, disambiguated corpora are needed to perform the training process , although there are a few approaches which work with raw corpora. Machine learning algorithms have been applied to learn classifiers from corpora in order to perform WSD, that is, algorithms are applied to certain features extracted from the annotated corpus and used to form a representation of each of the senses. This representation can then be applied to new instances in order to disambiguate them (Yarowsky, 1994; Ng, 1997; Escudero et al., 2000).

The last edition of the SENSEVAL competition has shown that corpus-based approaches achieve better results than knowledge-based ones. In the framework of corpus-based approaches, successful corpus-based approaches to POS tagging which used HMM have been extended in order to be applied to WSD. In (Segond et al., 1997), they estimated a bigram model of ambiguity classes from the *SemCor* corpus for the task of disambiguating a small set of semantic tags. Bigram models were also used in(Loupy et al., 1998). The task of sense disambiguating was carried out using the set of synsets of *WordNet* and using the *SemCor* corpus to train and to evaluate the system.

From all the precedent works and others, some conclusions could be established: sense disambiguation is a very difficult task and semantic resources to perform it are not sufficient. Despite these drawbacks, the good results ob-

[1] The *SemCor* corpus and *WordNet* are free available at http://www.cogsci.princeton.edu/˜wn/

[2] Information about the last edition of SENSEVAL can be found at http://www.sle.sharp.co.uk/senseval2/

tained by learning techniques in other disambiguation tasks and the preliminary results obtained in (Loupy et al., 1998), have encouraged us to present an approach to WSD based on HMM. A similar technique (Specialized HMM), which takes into account certain words to lexicalize the contextual language model, has been previously applied in order to solve POS tagging (Pla and Molina, 2001) and chunking (Molina and Pla, 2002) problems. In general, lexicalized HMMs perform better than non-lexicalized ones in these tasks.

The paper is organized as follows: in Section 2, we describe the WSD system proposed. In Section 3, we present the experimental work conducted on the *SemCor* corpus for the *all-words* task. Finally, we present some concluding remarks and future directions.

2. Description of the WSD system

We consider WSD to be a tagging problem which we propose to solve using a HMM formalism. Let $\mathcal{S}$ be the set of sense tags considered, and $\mathcal{W}$, the vocabulary of the application. From this point of view, tagging can be solved as a maximization problem. Given an input sentence, $W = w_1, \ldots, w_T$, where $w_i \in \mathcal{W}$, the tagging process consists of finding the sequence of senses ($S = s_1, \ldots, s_T$, where $s_i \in \mathcal{S}$) of maximum probability on the model, that is:

$$\begin{aligned}\widehat{S} &= \arg\max_S P(S|W) \\ &= \arg\max_S \left(\frac{P(S) \cdot P(W|S)}{P(W)}\right); S \in \mathcal{S}^T \qquad (1)\end{aligned}$$

Due to the fact that this maximization process is independent of the input sequence, and taking into account the Markov assumptions for a first-order HMM, the problem is reduced to solving the following equation:

$$\arg\max_S \left(\prod_{i:1\ldots T} P(s_i|s_{i-1}) \cdot P(w_i|s_i)\right) \qquad (2)$$

The parameters of equation 2 can be represented as a first-order HMM where each state corresponds to a sense s_i, where $P(s_i|s_{i-1})$ represent the transition probabilities between states and $P(w_i|s_i)$ represent the probability of emission of symbols, w_i, in every state, s_i. The parameters of this model are estimated by maximum likelihood from semantic annotated corpora using an appropriate smoothing method. Then, the semantic tagging is carried out using the Viterbi algorithm.

Starting from that general tagging scheme, we made certain decisions in order to improve the disambiguation process.

- We used certain resources, such as *WordNet* to know the possible semantic tags associated to the words. In addition, as we will show in the experimental section, we estimated the frequencies of each possible sense for a word from the *SemCor* corpus. This information is also available in *WordNet*.

- We decided which available input information is really relevant to the task. In this respect, we considered a concatenation of the lemma (l_i) and the POS[3] (p_i) associated to the word (w_i) as input vocabulary, if w_i has a sense in *WordNet*. For the words which do not have a sense in *WordNet*, we only consider their lemma (l_i) as input. So, in our HMM, $l_i \cdot p_i$ or l_i are the symbols emitted in the states.

  For example, for the input word *interest* which has an entry in *WordNet*, whose lemma is *interest* and whose POS is *NN*, the input considered in our system is *interest·1*. If the word does not have a sense in *WordNet*, such as the article *a*, we consider as input its lemma *a*.

- We defined the output semantic tag set by considering certain statistical information which was extracted from the annotated training set. In the *SemCor* corpus, each annotated word is tagged with a *sense_key* which has the form *lemma%lex_sense*. In general, we considered the *lex_sense* field of the *sense_key* associated to each lemma as the semantic tag in order to reduce the size of the output tag set. This does not lead to any loss of information because we can obtain the *sense_key* by concatenating the lemma to the output tag. For certain frequent lemmas, we considered a more fine-grained semantic tag: the *sense_key* or *synset*. These choices have been made experimentally by taking into account a set of frequent lemmas, $\mathcal{L}_s$, which were extracted from the training set.

  For instance, the input *interest·1* is tagged with the semantic tag *1:09:00::* in the training data set. If we estimate that the lemma *interest* belongs to $\mathcal{L}_s$, then the semantic tag is redefined as *interest·1:09:00::*.

  For the words without semantic information (tagged with the symbol *notag*), we have tested several transformations: to consider their POS in the states, to consider their lemma or to consider only one state for all these words. The approach that achieved the best results consisted of specializing the states with the lemma. For example, for the word *a* the output tag associated is *a·notag*.

The above decisions do not modify either the learning or the decoding process used. To apply them, we performed a transformation on the original training set to produce a new one which included these decisions (Molina and Pla, 2002). As we will show in the experimental results all these decisions improved the performance of our WSD system.

3. Experimental results

In order to evaluate the system proposed, we conducted some experiments on the *SemCor* corpus using *WordNet* 1.6 as a dictionary which supplies all possible semantic senses for a given word. In all the experiments performed, our system disambiguated all the polysemic lemmas, that is the coverage of our system was 100% (therefore, precision and recall were the same measure). We considered a lemma to

[3] We mapped the POS tags to the following tags: 1 for nouns, 2 for verbs, 3 for adjectives and 4 for adverbs.

be polysemic if it had more than one sense in *WordNet*, regardless of its POS. We reported the precision calculated on the polysemic words which are sense-tagged in the corpus.

We used the part of the *SemCor* corpus which is semantically annotated and supervised for nouns, verbs, adjectives and adverbs (that is, the files contained in Brown1 and Brown2 folders of *SemCor* corpus). The semantic tag set consists of 2,193 different senses which are included in *WordNet*. The corpus contains 414,228 tokens (359,732 word forms), 192,639 of these tokens have a semantic tag associated to them in the corpus and 162,662 are polysemic.

As we mentioned above, we selected a set of lemmas to specialize the states of the model. We conducted a tuning experiment on a development partition (10% of the corpus) to automatically extract the relevant lemmas for the model. This specialization criterion selected the lemmas whose frequency in the training data set was higher than a certain threshold. In order to determine which threshold maximized the performance of the model, we tuned it on the development partition using lemma sets of different sizes. For the Bigram model, the best performance was obtained by selecting the lemmas whose frequency was higher than 20 in the training data set (about 1600 lemmas). However, for the Unigram model, the best performance was obtaining using a total specialization, that is, using the *sense_keys* as semantic tags.

Once the set of frequent lemmas $\mathcal{L}_s$ was defined, we conducted a ten-fold cross validation experiment to evaluate our system. Each experimental partition consisted of 18 files taken from *SemCor* corpus as test data, and the rest as training data. The data test sets were completely different in the different partitions.

We considered a Baseline system which assigned the most frequent sense in the *SemCor* corpus given a lemma and its POS. This Baseline system achieved a precision of 70.79% which was calculated on the whole corpus. This result is very high, because the Baseline worked with a closed vocabulary. Due to this good result, we defined the emission probabilities of our models to be the probability distribution calculated on the entire *SemCor* corpus.

The results of the ten-fold cross validation are shown in Table 1. We compared the specialized models with respect to non-specialized ones. The basic unigram (UNI) and bigram (BIG) models are non-specialized models which took into account an input vocabulary that only consisted of lemmas. UNIpos and BIGpos are also non-specialized models whose input vocabulary considered the lemma and the POS as we mentioned in Section 2. These models (UNIpos and BIGpos) improved the performance of the basic models (UNI and BIG), showing that the POS information is important in differentiating among the different senses of a word. In addition, both Specialized models (UNIesp and BIGesp) outperformed the non-specialized ones. The Specialized Bigram model achieved a precision of 70.36%, which was slightly lower than the Baseline system precision.

4. Conclusions

In this paper, we proposed a word sense disambiguation system which is based on HMM and the use of *WordNet*.

Model	Precision
UNI	43.03%
UNIpos	54.06%
UNIesp	62.86%
BIG	65.49%
BIGpos	70.04%
BIGesp	70.36%
Baseline	70.79%

Table 1: Ten-fold cross validation precision results for polysemic words on the *SemCor* corpus.

We have made several versions of our WSD system. Firstly, we applied classic unigram and bigram models and, as we hoped, the bigram model outperformed the unigram model because the first one captures the context of the word to be disambiguated better. Secondly, we incorporated POS information to the input vocabulary which improved the performance and showed the relevance of this information in WSD. Finally, we specialized both the unigram and the bigram models in order to incorporate some relevant knowledge to the system. As we had also hope, specialized models improved the results of the non-specialized ones.

From the above experimentation, we concluded that the BIGesp model is the best model. However, when we compared its behavior against the Baseline system, which use the most frequent sense in *SemCor* corpus without any contextual information, we found that the Baseline system performs better than our best model (70.79% for Baseline and 70.36% for BIGesp).

The Baseline system only takes into account lexical information while our system also takes into account contextual information. However, the *SemCor* corpus as a training and evaluation resource for supervised sense taggers is somewhat limited, containing few tagged instances of the large majority of polysemic words. Therefore, corpus-based systems in general, and our approach in particular, could not estimate the parameters of the models sufficiently. Other approaches in the literature, that have also worked also on the *SemCor* corpus, did not improve the Baseline either. For example, in (Loupy et al., 1998), a system which offered one of the best results in Senseval-2 competition, the achieved precision on the *SemCor* corpus was very close to the Baseline precision.

A more objective analysis could be done on other corpora where the most frequent sense of polysemic words does not correspond to the most frequent sense in *WordNet*, which had been calculated on the *SemCor* corpus. Therefore, we are currently working on the application of our approach to WSD on other corpora.

5. Acknowledgments

This work has been supported by the Spanish research projects CICYT TIC2000–0664–C02–01 and TIC2000–1599–C01–01.

6. References

E. Agirre and G. Rigau. 1996. Word Sense Disambiguation Using Conceptual Density. In *Proceedings of the 16th*

International Conference on Computational Linguistics, COLING, Copenhagen, Denmark, August.

G. Escudero, L. Márquez, and G. Rigau. 2000. A comparison between supervised learning algorithms for Word Sense Disambiguation. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, September.

N. Ide and J. Véronis. 1998. Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, 24(1):1–40.

M. Lesk. 1986. Automated Sense Disambiguation using Machine-readable Dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC Conference*, pages 24–26, Toronto, Canada, June.

C. Loupy, M. El-Beze, and P. F. Marteau. 1998. Word Sense Disambiguation using HMM Tagger. In *Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC*, pages 1255–1258, Granada, Spain, May.

G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross, and K. J. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

G. A. Miller, M. Chodorow, S. Landes, C. Leacock, and R. G. Thomas. 1994. Using a Semantic Concordance for Sense Identificaction. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 240–243.

A. Molina and F. Pla. 2002. Shallow Parsing using Specialized HMMs. *Journal of Machine Learning Research*, 2:595–613.

H. T. Ng. 1997. Exemplar-Base Word Sense Disambiguation: Some Recent Improvements. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, EMNLP*.

F. Pla and A. Molina. 2001. Part-of-Speech Tagging with Lexicalized HMM. In *proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP2001)*, Tzigov Chark, Bulgaria, September.

P. S. Resnik and D. Yarowsky. 2000. Distinguishing systems and distinguishing senses: new evaluation methods for Word Sense Disambiguation. *Natural Language Engineering*, 6(3):113–133.

P. S. Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI*, pages 448–453, Montreal, Canada.

F. Segond, A. Schiller, G. Grefenstette, and J-P. Chanod. 1997. An Experiment in Semantic Tagging using Hidden Markov Model Tagging. In *Proceedings of the Joint ACL/EACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources*, pages 78–81, Madrid, Spain.

M. Stevenson and Y. Wilks. 2001. The Interaction of Knowledge Sources in Word Sense Disambiguation. *Computational Linguistics*, 27(3):321–349.

E. Voorhees. 1993. Using WordNet to Disambiguate Word Senses for Text Retrieval. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 171–180, Pittsburgh.

D. Yarowsky. 1992. Word-sense Disambiguations Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics, COLING*, pages 454–460, Nantes, France.

D. Yarowsky. 1994. Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 88–95, Las Cruces, NM. ACL.