A Hidden Markov Model Approach to Word Sense Disambiguation

Antonio Molina, Ferran Pla, and Encarna Segarra

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

Abstract. In this work, we propose a supervised approach to Word Sense Disambiguation which is based on Specialized Hidden Markov Models and the use of *WordNet*. Our approach formulates the disambiguation process as a tagging problem. The specialization process allows for the incorporation of additional knowledge into the models. We evaluated our system on the *English all-words* task of the *Senseval-2* competition. The performance of our system is in line with the best approaches for this task.

1 Introduction

Word Sense Disambiguation (WSD) consists of selecting the semantic sense of a word from all the possible senses given by a dictionary. It is well known that semantic information can be useful to solve different tasks such as parsing, machine translation, language understanding, information retrieval, etc. For example, a term-based information retrieval system answers the query *plants that live in the sea* with all the documents that contain the terms *plant*, *live* or *sea* regardless of their meaning. Some of these documents contain the term *plant* with the meaning "life form" and others contain the term *plant* with the meaning "factory". It would be interesting for the information retrieval system to give only the documents in which the term *plant* appears with the meaning "life form". To do this, the system should use a WSD module in order to obtain the correct sense of the ambiguous terms in the query.

WSD is a difficult task for various reasons. First, there is no consensus on the concept of sense, and consequently, different semantic tag sets can be defined. Second, the modeling of contextual dependencies is 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* [1] and the *SemCor* [1] corpus [2] are the most frequently used resources.

[1] The *SemCor* corpus and the knowledge base *WordNet* are freely available at http://www.cogsci.princeton.edu/~wn/

WordNet is a large-scale hand-crafted lexical knowledge base. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. *WordNet* provides all the possible senses for a certain word. These senses are sorted according to their frequency in the *SemCor* corpus. This corpus, which consists of 676,546 words, has been semantically annotated using the senses of *WordNet* and manually supervised.

There has been a wide range of approaches to the WSD problem (a detailed study can be found in [3] and [4]). 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*, a thesaurus, such as *Rodget's Thesaurus*, or large-scale hand-crafted knowledge bases, such as *WordNet* [5–10].

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. These algorithms extract certain features from the annotated corpus and use them to form a representation of each of the senses. This representation can then be applied to new instances in order to disambiguate them [11–13].

In the framework of corpus-based approaches, successful corpus-based approaches to POS tagging which used Hidden Markov Models (HMM) have been extended in order to be applied to WSD. In [14], 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 [15]. 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.

Senseval [2] competition can be viewed as the most important reference point for WSD. The last edition of this competition has shown that corpus-based approaches achieve better results than knowledge-based ones. Around 20 different systems participated in the *English all-words* task. The three best systems used supervised methods and they achieved a precision which ranked between 61.8% and 69.0%. The system *SMUaw* by Rada Mihalcea achieved the best precision (69.0%). It used a hybrid method which combined different knowledge sources: the *WordNet*, the *SemCor* corpus and a set of heuristics in order to obtain a set of sense-tagged word-word pairs. The second system in the competition (*CNTS-Antwerp*) by Veronique Hoste used a voting strategy in order to combine different learning algorithms, such as memory-based learning and rule induction. It used *SemCor* to train the different classifiers and obtained a precision of 63.6%. The *Sinequa-LIA-HMM* system by E. Crestan, M. El-Beze and C. Loupy achieved a

[2] Information about the latest edition of *Senseval* competition can be found at `http://www.sle.sharp.co.uk/senseval2/`

precision of 61.8%. It used a second-order HMM in a two-step strategy. First, it determined the semantic category associated to a word. Then, it assigned the most probable sense acording to the word and the semantic category. In addition, it used an alternative approach based on classification trees for certain words. The next two systems in the ranking by D. Fernández-Amorós (*UNED-AW-U*) used an unsupervised approach obtaining a precision of 55.0% and 56.9%. They constructed a relevance matrix from a large colecction of English books, which was used to filter the context of the words to be disambiguated. The rest of the systems, which are mainly based on unsupervised methods, gave a significantly lower performance than the methods mentioned above. Only a few of them gave a precision which was higher than 50%; however, they had a very low recall.

Some conclusions can be established from all these works. WSD is still an open problem in Natural Language Processing because the performance of the different systems is not satisfactory enough. In addition, the semantic resources available are not sufficient, because the number of senses is very large and annotated corpora do not have enough data to estimate appropriate models. Although this aspect specially affects corpus-based approaches, these achieve better results than knowledge-based ones.

In this paper we present a corpus-based approach to WSD based on Specialized HMM [16]. We chose this approach because it has been successfully applied to solve other Natural Language Processing problems such as Part-of-speech (POS) tagging [17] and chunking [18]. A preliminary evaluation of our WSD system was conducted on the *SemCor* corpus [19]. In that case, the precision results (70.39%) were not very satisfactory because our system performed much like the Baseline (we considered a system that assigned the most frequent sense in the *SemCor* corpus given a lemma and its POS as Baseline). In order to get a more objective analysis of the performance of our system, we selected the *English all-words* task of the *Senseval-2* competition.

The paper is organized as follows: in Sections 2 and 3, we describe the WSD system proposed and the learning process of the system. In Section 4, we present the experimental work conducted on the *Senseval-2 English all-words* task. Finally, we present some concluding remarks.

2 Description of the WSD System

We consider WSD to be a tagging problem which we propose to solve by using a HMM formalism. Figure 1 shows a scheme of the WSD system developed. The system has two components: *WordNet* and the HMM.

WordNet is used to know all the possible semantic tags associated to an input word. If the input word is unknown for the model (the word has not been seen in the training data set) the system takes the first sense provided by *WordNet*.

We can formulate the tagging process as a maximization problem. Let $\mathcal{S}$ be the set of sense tags considered, and $\mathcal{W}$, the vocabulary of the application. 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

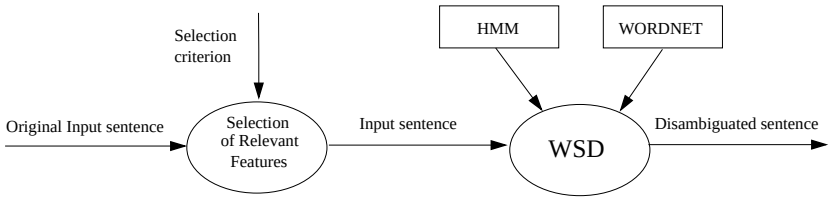

Fig. 1. System Description

probability on the model, that is:

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

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:

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

The parameters of equation 2 can be represented as a first-order HMM where each state corresponds to a sense s_i, $P(s_i|s_{i-1})$ 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.

Different kinds of available linguistic information can be useful to solve WSD. In particular, the available annotated corpora provide the following input features: words, lemmas and the corresponding part-of-speech tags. The first system step (see Figure 1) consists of applying a *selection criterion* to the original input sentence in order to choose the features which are relevant to the task. In our system, the vocabulary of the input sentence to the WSD module consists of the concatenation of its relevant features. The *selection criterion* is decided in the learning phase as we will show in Section 3.

Therefore, the disambiguation process is as follows: first, the original input sentence is processed in order to select its relevant features providing the input sentence; then, the semantic tagging is carried out through the Viterbi algorithm using the estimated HMM and *WordNet*.

3 Description of the Learning Phase

The learning process of a Specialized HMM [16, 18] is similar to the learning of a basic HMM. The only difference is that Specialized HMM are based on

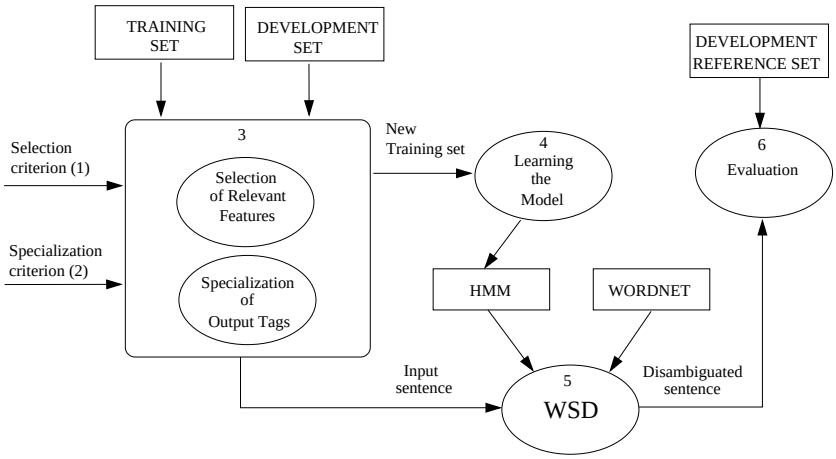

Fig. 2. Learning Phase Description

an appropriate definition of the input information to the learning process. This information consists of the input features (words, lemmas and POS tags) and the output tag set (senses) provided by the training corpus. A specialized HMM is built according to the following steps (see Figure 2):

1. To define which available input information is relevant to the task (*selection criterion*).
2. To define which input features are relevant to redefine or *specialize* the output tag set (*specialization criterion*). This specialization allows the model to better capture some restrictions relevant to the task.
3. To apply the chosen criteria to the original training data set to produce a new one.
4. To learn a model from the new training data set.
5. To disambiguate a development data set using that model.
6. To evaluate the output of the WSD system in order to compare the behavior of the selected criteria on the development set.

These steps are done using different combinations of the input information in order to determine the best selection and specialization criteria. From our experimental work, the best *selection criterion* consists of 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. Therefore, 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* and 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 its lemma *a* as input.

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

With respect to the *specialization criterion* we made the following decisions. 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 were 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 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*.

4 Experimental Results

We conducted some experiments on the *English all-words* task of the *Senseval-2* competition. This competition did not provide any training corpora for this task, so we used as training data the part of the *SemCor* corpus which is semantically annotated and supervised for nouns, verbs, adjectives and adverbs (that is, the files contained in the Brown1 and the 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. We used 10% of the training corpus as a development data set. The test data set provided by *Senseval-2* consisted of three Penn TreeBank documents which contained 2,473 sense-tagged words. POS information was extracted directly from the corresponding Penn TreeBank documents.

In the experiments, we used *WordNet* 1.6 as a dictionary which supplies all the possible semantic senses for a given word. Our system disambiguated all the polysemic lemmas, that is, the coverage of our system was 100% (therefore, precision[4] and recall [5] were the same). For unknown words (words that did not appear in the training data set), we assigned the first sense in *WordNet*.

We compared the following models. The unigram (UNI) and bigram (BIG) models are basic HMMs which take into account an input vocabulary that only consists of lemmas. UNIpos and BIGpos are models which were learnt using only

[4] precision = # of correctly disambiguated words / # of disambiguated words
[5] recall = # of correctly disambiguated words / # of words to be disambiguated

the best *selection criterion*. UNIesp and BIGesp are also learnt using this *selection criterion* and have been specialized using the best *specialization criterion*.

To build the specialized models, we selected the set of lemmas $\mathcal{L}_s$ beforehand. The *specialization criterion* consisted of selecting the lemmas whose frequency in the training data set was higher than a certain threshold (other specialization criteria could have been chosen, but frequency criterion usually worked well in other tasks as we reported in [18]). In order to determine which threshold maximized the performance of the model, we conducted a tuning experiment on the development set. For the BIGesp model, the best performance was obtained using a threshold of 20 ($|\mathcal{L}_s|$ was about 1,600 lemmas).

Table 1. Precision results for the *English all-words* task in *Senseval-2*. 100% of the words were tagged.

Model	Precision
UNI	40.00%
UNIpos	52.30%
UNIesp	58.80%
BIG	50.10%
BIGpos	58.20%
BIGesp	60.20%

The results for the *English all-words* task are shown in Table 1. The UNIpos and BIGpos models 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 best performance was achieved by the Specialized Bigram model (BIGesp) with a precision of 60.20%, which was slightly higher than the Baseline precision (58.0%). This result is in line with the results provided for the best systems in *Senseval-2*. This comfirms that Specialized HMMs can be applied to WSD, as successfully as they have been applied to other disambiguation tasks. The most similar approach to our system (Sinequa-LIA-HMM) achieved a result which was slightly better (61.8%), but it combined the HMM model with a classification tree method to disambiguate some selected words.

5 Conclusions

In this paper, we have proposed a Word Sense Disambiguation system which is based on HMM and the use of *WordNet*. We made several versions of our WSD system. Firstly, we applied classic unigram and bigram models and, as we had expected, the bigram model outperformed the unigram model. This is because the bigram model better captures the context of the word to be disambiguated. 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. Again, as we expected, specialized models improved the results of the non-specialized ones.

From the above experimentation, we conclude that the BIGesp model is the best model. This model gave a precision of 60.20% in the *English all-words* task of the *Senseval-2* competition, which was only outperformed by the three best systems for the same task. This is a good result taking into account that our WSD system is mainly an adaptation of our POS tagging and chunking systems. This adaptation consists of an appropriate definition of the relevant input information and the output tag set.

Finally, we think that we could improve our WSD system through a more adequate definition of the selection and specialization criteria. To do this, a larger training data set, which is close to the task, would be necessary. Moreover, this definition could be done using linguistic knowledge as well.

6 Acknowledgments

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

References

1. Miller, G.A., Beckwith, R., Fellbaum, C.D., Gross, D., Miller, K.J.: WordNet: An on-line lexical database. International Journal of Lexicography **3** (1990) 235–244
2. Miller, G.A., Chodorow, M., Landes, S., Leacock, C., Thomas, R.G.: Using a Semantic Concordance for Sense Identificaction. In: Proceedings of the ARPA Workshop on Human Language Technology. (1994) 240–243
3. Ide, N., Véronis, J.: Word Sense Disambiguation: The State of the Art. Computational Linguistics **24** (1998) 1–40
4. Resnik, P., Yarowsky, D.: Distinguishing systems and distinguishing senses: new evaluation methods for Word Sense Disambiguation. Natural Language Engineering **6** (2000) 113–133
5. Lesk, M.: 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, Toronto, Canada (1986) 24–26
6. Yarowsky, D.: 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, Nantes, France (1992) 454–460
7. Voorhees, E.: 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, Pittsburgh (1993) 171–180
8. Resnik, P.S.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI, Montreal, Canada (1995) 448–453

9. Agirre, E., Rigau, G.: Word Sense Disambiguation Using Conceptual Density. In: Proceedings of the 16th International Conference on Computational Linguistics, COLING, Copenhagen, Denmark (1996)
10. Stevenson, M., Wilks, Y.: The Interaction of Knowledge Sources in Word Sense Disambiguation. Computational Linguistics **27** (2001) 321–349
11. Yarowsky, D.: 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, Las Cruces, NM, ACL (1994) 88–95
12. Ng, H.T.: Exemplar-Base Word Sense Disambiguation: Some Recent Improvements. In: Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, EMNLP. (1997)
13. Escudero, G., Márquez, L., Rigau, G.: A comparison between supervised learning algorithms for Word Sense Disambiguation. In: Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal (2000)
14. Segond, F., Schiller, A., Grefenstette, G., Chanod, J.P.: 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, Madrid, Spain (1997) 78–81
15. Loupy, C., El-Beze, M., Marteau, P.F.: Word Sense Disambiguation using HMM Tagger. In: Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC, Granada, Spain (1998) 1255–1258
16. Molina, A., Pla, F., Segarra, E.: Una formulación unificada para resolver distinto problemas de ambigüedad en PLN. Revista para el Procesamiento del Lenguaje Natural (to be published) (2002)
17. Pla, F., Molina, A.: Part-of-Speech Tagging with Lexicalized HMM. In: proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP2001), Tzigov Chark, Bulgaria (2001)
18. Molina, A., Pla, F.: Shallow Parsing using Specialized HMMs. Journal of Machine Learning Research **2** (2002) 595–613
19. Molina, A., Pla, F., Segarra, E., Moreno, L.: Word Sense Disambiguation using Statistical Models and WordNet. In: Proceedings of 3rd International Conference on Language Resources and Evaluation, LREC2002, Las Palmas de Gran Canaria, Spain (2002)