SENSEVAL-3: Third International Workshop on the Evaluation of Systems
for the Semantic Analysis of Text, Barcelona, Spain, July 2004
Association for Computational Linguistics

WSD system based on Specialized Hidden Markov Model (upv-shmm-eaw)

Antonio Molina, Ferran Pla and **Encarna Segarra**
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n València (Spain)
{amolina,fpla,esegarra}@dsic.upv.es

Abstract

We present a supervised approach to Word Sense Disambiguation (WSD) based on Specialized Hidden Markov Models. We used as training data the Semcor corpus and the test data set provided by Senseval 2 competition and as dictionary the Wordnet 1.6. We evaluated our system on the English all-word task of the Senseval-3 competition.

1 Description of the WSD System

We consider WSD to be a tagging problem (Molina et al., 2002a). The tagging process can be formulated as a maximization problem using the Hidden Markov Model (HMM) formalism. Let $\mathcal{O}$ be the set of output tags considered, and $\mathcal{I}$, the input vocabulary of the application. Given an input sentence, $I = i_1, \ldots, i_T$, where $i_j \in \mathcal{I}$, the tagging process consists of finding the sequence of tags ($O = o_1, \ldots, o_T$, where $o_j \in \mathcal{O}$) of maximum probability on the model, that is:

$$\widehat{O} = \arg\max_{O} P(O|I) = \arg\max_{O} \left(\frac{P(O) \cdot P(I|O)}{P(I)} \right); O \in \mathcal{O}^T \quad (1)$$

Due to the fact that the probability $P(I)$ is a constant that can be ignored in the maximization process, the problem is reduced to maximize the numerator of equation 1. To solve this equation, the Markov assumptions should be made in order to simplify the problem. For a first-order HMM, the problem is reduced to solve the following equation:

$$\arg\max_{O} \left(\prod_{j:1..T} P(o_j|o_{j-1}) \cdot P(i_j|o_j) \right) \quad (2)$$

The parameters of equation 2 can be represented as a first-order HMM where each state corresponds to an output tag o_j, $P(o_j|o_{j-1})$ represent the transition probabilities between states and $P(i_j|o_j)$ represent the probability of emission of input symbols, i_j, in every state, o_j. The parameters of this model are estimated by maximum likelihood from semantic annotated corpora using an appropriate smoothing method (linear interpolation in our work).

Different kinds of available linguistic information can be useful to solve WSD. The training corpus we used provides as input features: words ($\mathcal{W}$), lemmas ($\mathcal{L}$) and the corresponding POS tags ($\mathcal{P}$); and it also provides as output tags the *WordNet* senses.

WordNet senses can be represented by a *sense_key* which has the form *lemma%lex_sense*. The high number of different *sense_keys* and the scarce annotated training data make difficult the estimation of the models. In order to alleviate this sparness problem we considered the *lex_sense* field ($\mathcal{S}$) of the *sense_key* associated to each lemma as the semantic tag. This assumption reduces the size of the output tag set and it does not lead to any loss of information because we can obtain the *sense_key* by concatenating the lemma to the output tag.

Therefore, in our system the input vocabulary is $\mathcal{I} = \mathcal{W} \times \mathcal{L} \times \mathcal{P}$, and the output vocabulary is $\mathcal{O} = \mathcal{S}$. In order to incorporate this kind of information to the model we used Specialized HMM (SHMM) (Molina et al., 2002b). This technique has been successfully applied to other disambiguation tasks such as part-of-speech tagging (Pla and Molina, 2004) and shallow parsing (Molina and Pla, 2002).

Other HMM-based approaches have also been 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 the semantic categories corresponding to the lexicographer level. These semantic categories are codified into the *lex_sense* field. A second-order HMM was used in (Loupy et al., 1998) in a two-step strategy. First, they determined the semantic category associated to a word. Then, they assigned the most probable sense according to the word and the semantic category.

A SHMM consists of changing the topology of the HMM in order to get a more accurate model

which includes more information. This is done by means of an initial step previous to the learning process. It consists of the redefinition of the input vocabulary and the output tags. This redefinition is done by means of two processes which transform the training set: the *selection* process, which is applied to the input vocabulary, and the *specialization* process, which redefines the output tags.

1.1 Selection process

The aim of the *selection* process is to choose which input features are relevant to the task. This process applies a determined *selection criterion* to $\mathcal{I}$ that produces a new input vocabulary ($\widetilde{\mathcal{I}}$). This new vocabulary consists of the concatenation of the relevant input features selected.

Taking into account the input vocabulary $\mathcal{I} = \mathcal{W} \times \mathcal{L} \times \mathcal{P}$, some selection criteria could be as follows: to consider only the word (w_i), to consider only the lemma (l_i), to consider the concatenation of the word and its POS[1] ($w_i \cdot p_i$), and to consider the concatenation of the lemma and its POS ($l_i \cdot p_i$). Moreover, different criteria can be applied depending on the kind of word (e.g. distinguishing content and non-content words).

For example, for the input word *interest*, which has an entry in *WordNet* and whose lemma and POS are *interest* and *NN* (common noun) respectively, the input considered could be *interest·1*. For a non-content word, such as the article *a*, we could consider only its lemma *a* as input.

1.2 Specialization process

The *specialization process* allows for the codification of certain information into the context (that is, into the states of the model). It consists of redefining the output tag set by adding information from the input. This redefinition produces some changes in the model topology, in order to allow the model to better capture some contextual restrictions and to get a more accurate model.

The application of a *specialization criterion* to $\mathcal{O}$ produces a new output tag set ($\widetilde{\mathcal{O}}$), whose elements are the result of the concatenation of some relevant input features to the original output tags.

Taking into account that the POS input feature is already codified in the *lex_sense* field, only words or lemmas can be considered in the specialization process ($w_i \cdot$ lex_sense$_i$ or $l_i \cdot$ lex_sense$_i$).

This specialization can be *total* or *partial* depending on whether we specialize the model with all the elements of a feature or only with a subset of them.

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

For instance, the input token *interest·1* is tagged with the semantic tag *1:09:00::* in the training data set. If we estimate that the lemma *interest* should specialize the model, then the semantic tag is redefined as *interest·1:09:00::*. Non-content words, that share the same output tag (the symbol *notag* in our system), could be also considered to specialize the model. For example, for the word *a*, the specialized output tag associated could be *a·notag*.

1.3 System scheme

The disambiguation process is presented in (Figure 1). First, the original input sentence (I) is processed in order to select its relevant features, providing the input sentence ($\widetilde{I}$). Then, the semantic tagging is carried out through the Viterbi algorithm using the estimated SHMM. *WordNet* is used to know all the possible semantic tags associated to an input word. If the input word is unknown for the model (i.e., the word has not been seen in the training data set) the system takes the first sense provided by *WordNet*.

The learning process of a SHMM is similar to the learning of a basic HMM. The only difference is that SHMM are based on 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 SHMM 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*).
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 input features in order to determine the best *selection* criterion and the best total *specialization* criterion. Once these criteria are determined, some partial specializations are tested in order to improve the performance of the model.

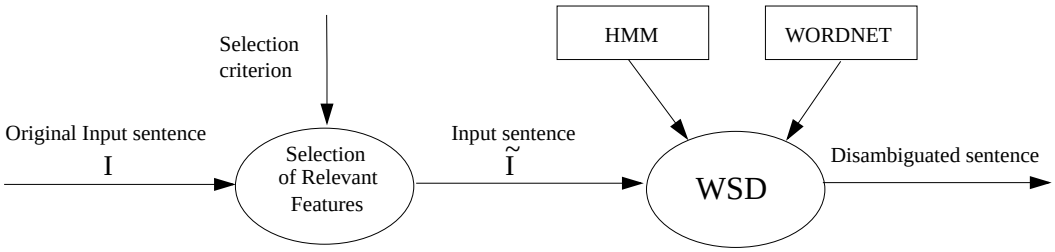

Figure 1: System Description

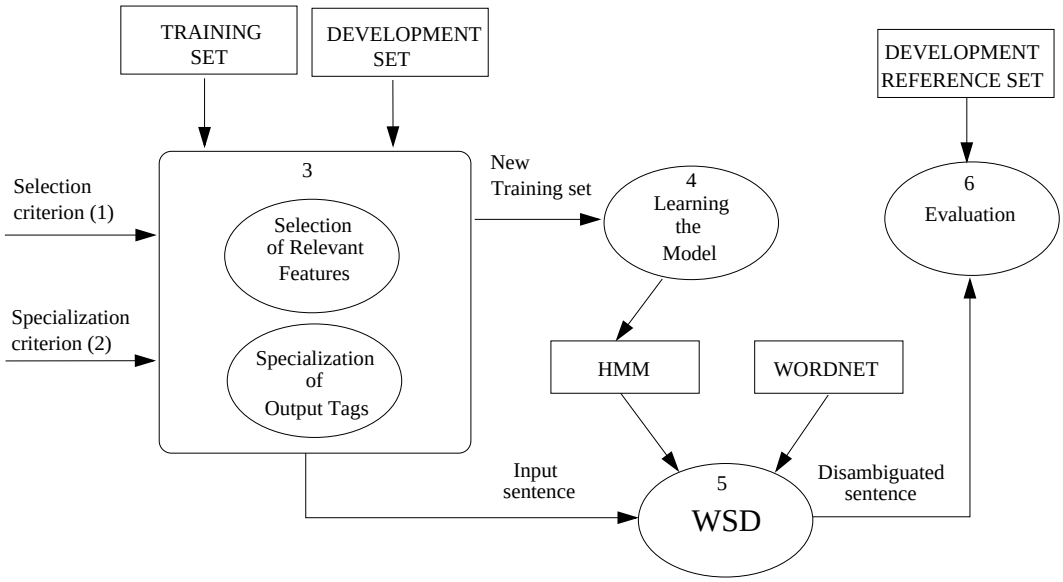

Figure 2: Learning Phase Description

2 Experimental Work

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), and the test data set provided by *Senseval-2*. We used 10% of the training corpus as a development data set in order to determine the best *selection* and *specialization* criteria.

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 and recall were the same). For unknown words (words that did not appear in the training data set), we assigned the first sense in *WordNet*.

The best *selection criterion* determined from the experimental work on the development set is as follows: if a word w_i has a sense in *WordNet* we concatenate the lemma (l_i) and the POS (p_i) associated to the word (w_i) as input vocabulary. For non-content words, we only consider their lemma (l_i) as input.

The best *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 (Molina and Pla, 2002)). In order to determine which threshold maximized the performance of the model, we conducted a tuning experiment on the development set. The best performance was obtained using the lemmas whose frequency was higher than 20 (about 1,600 lemmas).

The performance of our system on the Senseval 3 data test set was 60.9% of precision and recall.

3 Concluding remarks

In our WSD system, the choice of the best *specialization criterion* is based on the results of the system on the development set. The tuning experiments included totally specialized models, which is equivalent to consider the *sense_keys* as the output vocab-

ulary, non-specialized models, which is equivalent to consider the *lex_senses* as the output vocabulary, and partially specialized models using different sets of lemmas.

For the best *specialization criterion*, we have not studied the linguistic characteristics of the different groups of *synsets* associated to the same *lex_sense* for non-specialized output tags. We think that we could improve our WSD system through a more adequate definition of the *selection* and *specialization* criteria. This definition could be done using semantic knowledge about the domain of the task.

4 Acknowledgments

This work has been supported by the Spanish research projects CICYT TIC2003-07158-C04-03 and TIC2003-08681-C02-02.

References

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.

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

Antonio Molina, Ferran Pla, and Encarna Segarra. 2002a. A Hidden Markov Model Approach to Word Sense Disambiguation. In *Proceedings of the VIII Conferencia Iberoamericana de Inteligencia Artificial, IBERAMIA2002*, Sevilla, Spain.

Antonio Molina, Ferran Pla, and Encarna Segarra. 2002b. Una formulaci´on unificada para resolver distinto problemas de ambigüedad en PLN. *Revista para el Procesamiento del Lenguaje Natural, (SEPLN'02)*, Septiembre.

Ferran Pla and Antonio Molina. 2004. Improving Part-of-Speech Tagging using Lexicalized HMMs. *Natural Language Engineering*, 10. In press.

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.