Pattern Learning and Active Feature Selection for Word Sense Disambiguation

Rada F. MIHALCEA
Southern Methodist University
Dallas, Texas, 75275-0122
rada@seas.smu.edu

Dan I. MOLDOVAN
University of Texas at Dallas
Richardson, Texas, 75083-0688
moldovan@utdallas.edu

Abstract

We present here the main ideas of the algorithm employed in the *SMUls* and *SMUaw* systems. These systems have participated in the SENSEVAL-2 competition attaining the best performance for both English all words and English lexical sample tasks[1]. The algorithm has two main components (1) pattern learning from available sense tagged corpora (SemCor) and dictionary definitions (WordNet), and (2) instance based learning with active feature selection, when training data is available for a particular word.

1 Introduction

It is well known that WSD constitutes one of the hardest problems in Natural Language Processing, yet is a necessary step in a large range of applications including machine translation, knowledge acquisition, coreference, information retrieval and others. This motivates a continuously increasing number of researchers to develop WSD systems and devote time to finding solutions for this challenging problem.

The system presented here was initially designed for the semantic disambiguation of *all words* in open text. The SENSEVAL competitions created a good environment for supervised systems and this encouraged us to improve our system with the capability of incorporating larger training data sets when provided.

There are two important modules in this system. The first one uses pattern learning relying on large sense tagged corpora to tag all words in open text. The second module is triggered only for the words with large training data, as was the case with the words from the lexical sample tasks. It uses an instance based learning algorithm with active feature selection.

To our knowledge, both pattern learning and active feature selection are novel approaches in the WSD field, and they led to very good results during the SENSEVAL-2 evaluation exercise.

2 System description

The WSD algorithm used in this system has the capability of tagging words when no specific sense tagged corpora is available, automatically scaling up to larger training data[2] when provided.

Due to space constraints, we will not be able to give a detailed description of the system. However we try to gain space and replace one thousand words with a picture: Figure 1 shows an overview of the system architecture. It illustrates the two main components, namely pattern learning from available sense tagged corpora and dictionary definitions and instance based learning with active feature selection. The two modules are preceded by a preprocessing phase which includes compound concept identification, and followed by a default phase that assigns the most frequent sense as a last resort, when no other previous methods could be applied. The shaded areas in Figure 1 are specific for the case when larger training data sets are available.

During the preprocessing stage, SGML tags are eliminated, the text is tokenized, part of speech tags are assigned using Brill tagger (Brill, 1995), and Named Entities (NE) are identified with an *in-house* implementation of an NE recognizer. To identify collocations, we determine sequences of words that form compound concepts defined in WordNet.

In the second step, patterns[3] are learned from WordNet, SemCor and GenCor, which is a large

[1] This is in conformity with the original ranking, following the evaluation of systems answers submitted before deadline.

[2] I.e. in addition to the publicly available sense tagged corpora

[3] We alternatively call them *rules* as they basically specify the sense triggered by a given local context, using rules like "if the word before is X then sense is Y"

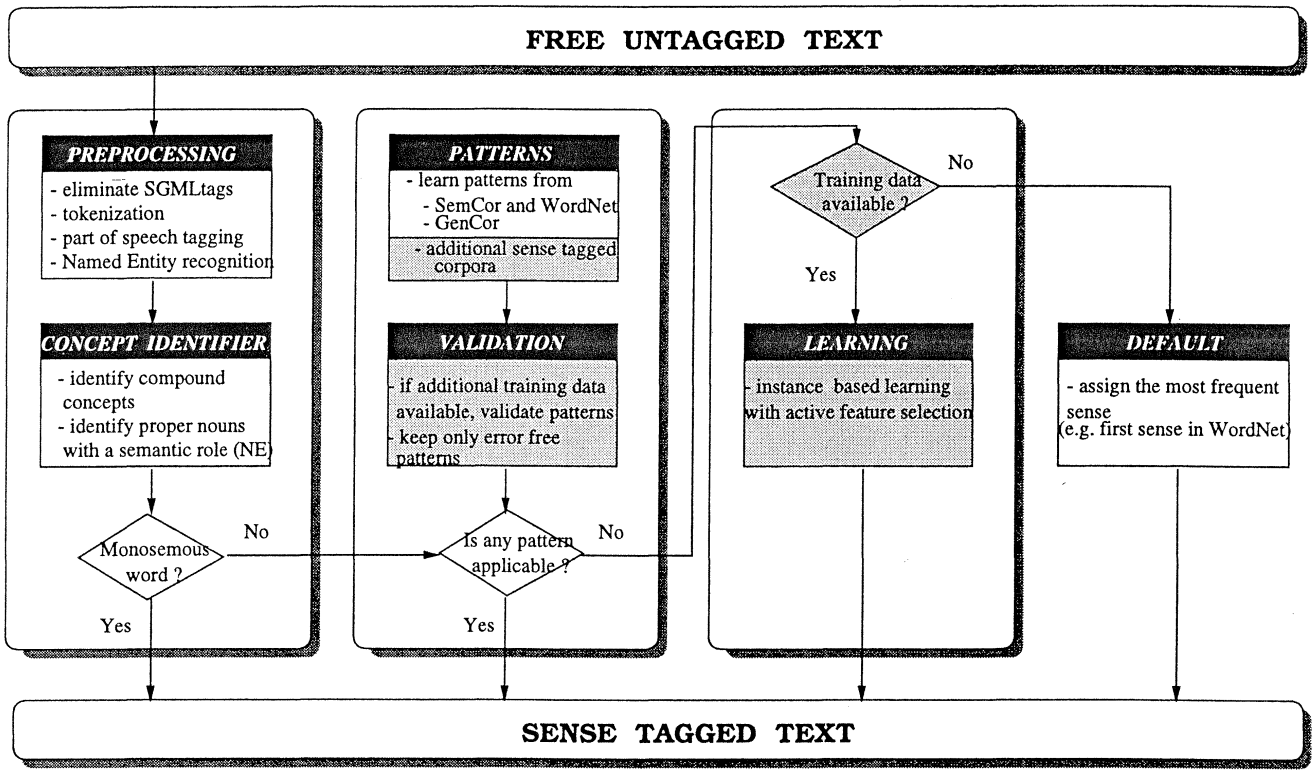

Figure 1: System architecture

sense tagged corpus automatically built via a set of heuristics. If additional training data is available, patterns are filtered through a validation process. Practically, the patterns are applied on the sense tagged data and we keep only those with no counter-examples found in the training sets.

The third step consists of a learning mechanism with active feature selection. This step is initiated only for those words with a sufficiently large number of examples, as was the case with the words in the SENSEVAL lexical sample tasks.

3 Pattern learning

This module is intended for solving the semantic ambiguity of *all* words in open text. To this end, we build disambiguation patterns using SemCor, WordNet and GenCor. Several processing steps were required to transform the first two resources into a useful corpus for our task. Moreover, these lexical resources coupled with a set of heuristics were used as seeds for generating a new sense tagged corpus called GenCor.

SemCor The SENSEVAL-2 English tasks have decided to use the WordNet 1.7 sense inventory, and therefore we had to deal with the task of mapping SemCor senses, which were assigned using an earlier version of WordNet, to the corresponding senses in WordNet 1.7. When a word sense from WordNet 1.6 is missing we assign a default sense of 0.[4]

WordNet The main idea in generating a sense tagged corpus out of WordNet is very simple. It is based on the underlying assumption that each example pertains to a word belonging to the current synset, thereby allowing us to assign the correct sense to at least one word in each example. For instance, the example given for *mother#4* is *"necessity is the mother of invention"*, and the word *mother* can be tagged with its appropriate sense.

GenCor is a generated sense tagged corpus, containing at the moment about 160,000 tagged words, which uses as seeds the sense tagged examples from SemCor and WordNet, as well as some of the principles for generating sense tagged corpora presented in (Mihalcea and Moldovan, 1999). Due to space limitations we cannot present here the methodology for creating this corpus. A thorough description is provided in (Mihalcea, 2001).

[4]SemCor 1.7a is available for download at http://www.seas.smu.edu/~rada/semcor

Once we have created this large corpus with examples of word meanings, we can start to learn patterns. A pattern basically consists of the local context for each semantically tagged word found in the corpus. The local context is formed by a window of N words to the left and M words to the right of each word considered. Additionally, a set of constraints is applied to filter out meaningless patterns.

Patterns are formed following the rules for regular expressions. Each word in the corpus is represented by its base form, its part of speech, its sense, if there is any provided, and its hypernym, again if the sense is known. Any of these word components can be unspecified, and therefore denoted with the symbol *. A count is also associated with every pattern, indicating the number of times it occurred in the corpus.

When trying to disambiguate a word, first we search for all available patterns that match the current context. In doing so, we use the current word as a pivot to perform matching. If there are several patterns available, then the decision of which pattern to apply is based on the pattern *strength*. The strength of a pattern is evaluated in terms of (1) number of specified components, (2) number of occurrences and (3) pattern length.

<the/DT modal/JJ/1 age/NN at/IN> is considered to be stronger than <modal/NN/1 age/NN>. Also, <clear/JJ/4 water/NN/1> is stronger than <clear/JJ water/NN/1>. Moreover, the inclusion of the hypernym among the word components gives us the means for generalization. For instance, <*/NN/*/room/1 door/NN/1> matches "kitchen door" as well as "bedroom door".

Another important step performed during the all words disambiguation task is sense propagation. The patterns do not guarantee a complete coverage of all words in input text, and therefore additional methods are required. We use a cache-like procedure which assigns to each ambiguous word the sense of its closest occurrence, if any can be found. The words still ambiguous at this point are assigned by default the first sense in WordNet.

Words with a significant number of semantic tagged examples constitute a special case in our system. There is a second module designed to handle the semantic disambiguation of these words. This module, described in the following section, exploits the benefits of having larger training data available for a particular word.

4 Learning with active feature selection

Learning mechanisms for disambiguating word senses have a long tradition in the WSD field. For our system, we have decided for an instance based algorithm with information gain feature weighting. The reasons for this decision are three fold: first, it has been advocated that forgetting exceptions is harmful for language learning applications (Daelemans et al., 1999), and instance based algorithms are known for their property of taking into consideration every single training example when making a classification decision; secondly, instance based learning algorithms have been successfully used in WSD applications (Veenstra et al., 2000); finally, this type of algorithms are efficient in terms of training and testing time. We have initially used the MLC++ implementation, and later on switched to Timbl (Daelemans et al., 2001).

Even more important than the choice of learning methodology is the selection of features employed during the learning process. There are several features recognized as good indicators of word sense, including the word itself (CW) and its part of speech (CP), surrounding words and their parts of speech (CF), collocations (COL), syntactic roles, keywords in contexts (SK). More recently, other possible features have been investigated: bigrams in context (B), named entities (NE), the semantic relation with the other words in context, etc.

Our intuition was that different sets of features have different effects depending on the ambiguous word considered. Feature weighting was clearly proven to be an advantageous approach for a large range of applications, including WSD. Still, weights are computed independently for each feature and therefore this strategy does not always guarantee to provide the best results.

For our system, we actively select features using a forward search algorithm. In this way, we practically generate *meta word experts*. Each word will have a different set of features that will eventually lead to the best disambiguation accuracy.

Using this approach, we combine the advantages of instance based learning mechanisms that have the nice property of *"not forgetting exceptions"*, with an optimized feature selection scheme. One could argue that decision trees have the capability of selecting relevant features, but

it has been shown (Almuallim and Dietterich, 1991) that irrelevant features significantly affect the performance of decision trees as well.

The algorithm for active feature selection is sketched in Figure 2. It is worth mentioning that in step 2, the training and testing corpora are extracted for each ambiguous word. This means that examples pertaining to the word *"dress down"* are separated from the examples for the single word *"dress"*.

1. *Generate pool of features PF = $\{F_i\}$. Initialize the set of selected features with the empty set SF=$\{\emptyset\}$.*
2. *Extract training and testing corpora for the given target ambiguous word.*
3. *For each feature F_i in the pool PF:*
   - 3.1. *Run a 10-fold cross validation on the training set; each example in the training set contains the features in SF* **and** *the feature F_i.*
   - 3.2. *Determine the feature F_i leading to the best accuracy.*
   - 3.3. *Remove F_i from PF and add it to SF.*
4. *Repeat step 3 until no improvements are obtained.*

Figure 2: Algorithm for active feature selection

The pool PF contains a large number of features, including those previously mentioned CW, CP, CF, COL, SK, B, NE, as well as other features like the noun before and after (NB, NA), head of the noun phrase, surrounding verbs, and others.

5 Results in SENSEVAL-2

The overall performance of the system in the English all words task was 69% for fine-grained scoring, respectively 69.8% for coarse-grained scoring (*SMUaw*). In the English lexical sample task, we obtained 63.8% for fine-grained scoring, respectively 71.2% for coarse-grained scoring (*SMUls*). These results ranked our system before deadline as the best performing for both tasks.

Discussion

There were several interesting cases encountered in the SENSEVAL-2 data, justifying our approach of using *active* feature selection. The influence of a feature greatly depends on the target word: a feature can increase the precision for a word, while making things worse for another word. For example, a word such as *free* does not benefit from the surrounding keywords (SK) feature, whereas *colourless* gains almost 7% in precision when this feature is used.

free.a[CW CP CF SK]	→	57.85%
free.a[CW CP CF]	→	63.57%
colourless.a[CW CP CF]	→	78.57%
colourless.a[CW CP CF SK]	→	85.71%

Another interesting example is constituted by the noun *chair*, which was disambiguated with high precision by simply using the current word (CW) feature. This is explained by the fact that the most frequent senses are **C***hair* meaning *person* and **c***hair* meaning *furniture*, and therefore the distinction between lower and upper case spellings makes the distinction among the different meanings of this word.

We have also tested the system on the SENSEVAL-1 data, and performed the disambiguation task in respect with Hector definitions, as required by the first disambiguation exercise. The overall result achieved on this data was higher than the one reported by the best performing system. Besides proving the validity of our approach, this fact also proved that our system is not tight in any ways to the sense inventory or data format employed.

6 Conclusion

Pattern learning and active feature selection are new approaches in the WSD field. They have been implemented in a system that participated in the SENSEVAL-2 competition, with an excellent performance in both *English all words* and *English lexical sample* tasks.

References

H. Almuallim and T.G. Dietterich. 1991. Learning with many irrelevant features. In *Proceedings of AAAI-91*, volume 2, pages 547–552, Anaheim, California.

E. Brill. 1995. Transformation-based error driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–566, December.

W. Daelemans, A. van den Bosch, and J. Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3):11–34.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2001. Timbl: Tilburg memory based learner, version 4.0, reference guide. Technical report, University of Antwerp.

R. Mihalcea and D.I. Moldovan. 1999. An automatic method for generating sense tagged corpora. In *Proceedings of AAAI-99*, Orlando, FL, July.

R. Mihalcea. 2001. GenCor: a large semantically tagged corpus. (in preparation).

J. Veenstra, A. van den Bosch, S. Buchholz, W. Daelemans, and J. Zavrel. 2000. Memory-based word sense disambiguation. *Computers and the Humanities*, 34:171–177.