LCC-WSD: System Description for English Coarse Grained All Words Task at SemEval 2007

Adrian Novischi, Munirathnam Srikanth and Andrew Bennett
Language Computer Corp.
Richardson, TX
{adrian,srikanth,abennet}@languagecomputer.com

Abstract

This document describes the Word Sense Disambiguation system used by Language Computer Corporation at English Coarse Grained All Word Task at SemEval 2007. The system is based on two supervised machine learning algorithms: Maximum Entropy and Support Vector Machines. These algorithms were trained on a corpus created from SemCor, Senseval 2 and 3 all words and lexical sample corpora and Open Mind Word Expert 1.0 corpus. We used topical, syntactic and semantic features. Some semantic features were created using WordNet glosses with semantic relations tagged manually and automatically as part of eXtended WordNet project. We also tried to create more training instances from the disambiguated WordNet glosses found in XWN project (XWN, 2003). For words for which we could not build a sense classifier, we used First Sense in WordNet as a back-off strategy in order to have coverage of 100%. The precision and recall of the overall system is 81.446% placing it in the top 5 systems.

1 Introduction

The performance of a Word Sense Disambiguation (WSD) system using a finite set of senses depends greatly on the definition of the word senses. Fine grained senses are hard to distinguish while coarse grained senses tend to be more clear. Word Sense Disambiguation is not a final goal, but it is an intermediary step used in other Natural Processing applications like detection of Semantic Relations, Information Retrieval or Machine Translation. Word Sense Disambiguation is not useful if it is not performed with high accuracy (Sanderson, 1994). A coarse grained set of sense gives the opportunity to make more precise sense distinction and to make a Word Sense Disambiguation system more useful to other tasks.

Our goal at SemEval 2007 was to measure the performance of known supervised machine learning algorithm using coarse grained senses. The idea of using supervised machine learning for WSD is not new and was used for example in (Ng and Lee, 1996). We made experiments with two supervised methods: Maximum Entropy (ME) and Support Vector Machines (SVM). These supervised algorithms were used with topical, syntactic and semantic features. We trained a classifier for each word using both supervised algorithms. New features were added in 3 incremental steps. After an initial set of experiments the algorithm performance was enhanced using a greedy feature selection algorithm similar to one in (Mihalcea, 2002). In order to increase the number of training instances, we tried to use the disambiguated WordNet glosses from XWN project (XWN, 2003). Combining other corpora with disambiguated glosses from XWN did not provide any improvement so we used XWN as a fall back strategy for 70 words that did not have any training examples in other corpora but XWN.

Section 2 describes the supervised methods used by our WSD system, the pre-processing module and the set of features. Section 3 presents the experiments we performed and their results. Section 4 draws the conclusions.

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 223–226,
Prague, June 2007. ©2007 Association for Computational Linguistics

2 System Description

The system contains a preprocessing module used before computing the values of the features needed by the machine learning classifiers. The preprocessing module perform the following steps:

- Tokenization: using an in house text tokenizer
- Named Entity Recognition: using an in house system
- Part of Speech Tagging: normally we use the Brill tagger, but we took advantage of the part of speech tags given in the test file
- WordNet look-up to check if the word exists in WordNet and to get its lemma, possible part of speech for that lemma and if the word has a single sense or not. For SemEval English Coarse All Words task we took advantage by the lemma provided in the test file.
- Compound concept detection: using a classifier based on WordNet
- Syntactic Parsing: using an in-house implementation of Collin's parser (Glaysher and Moldovan, 2006)

The Maximum Entropy classifier is a C++ implementation found on web (Le, 2006). The classifier was adapted to accept symbolic features for classification tasks in Natural Language Processing.

For training SVM classifiers we used LIBSVM package (Chang and Lin, 2001). Each symbolic feature can have a single value from a finite set of values or can be assigned a subset of values from the set of all possible values. For each value we created a mapping between the feature value and a dimension in the N-dimensional classification space and we assigned the number 1.0 to that dimension if the feature had the corresponding value or 0.0 otherwise.

We first performed experiments with our existing set of features used at Senseval 3 All Words task. We call this set FS_1. Then we made three incremental changes to improve the performance.

The initial set contains the following features: current word form (CRT_WORD) and part of speech (CRT_POS), contextual features (CTX_WORD) in a window (-3,3) words, collocations in a window of (-3,3) words (COL_WORD), keywords (KEYWORDS) and bigrams (BIGRAMS) in a window of (-3,3) sentences, verb mode (VERB_MODE) which can take 4 values: ACTIVE, INFINITIVE, PAST, GERUND, verb voice (VERB_VOICE) which can take 2 values ACTIVE, PASSIVE, the parent of the current verb in the parse tree (CRT_PARENT) (ex: VP, NP), the first ancestor that is not VP in the parse tree (RAND_PARENT) (like S, NP, PP, SBAR) and a boolean flag indicating if the current verb belongs to the main clause or not (MAIN_CLAUSE).

We added new features to the initial set. We call this set FS_2.

- The lemmas of the contextual words in the window of (-3, 3) words around the target word (CTX_LEMMA).
- Collocations formed with the lemma of surrounding words in a window of (-3, 3) (COL_LEMMA)
- The parent of the contextual words in the parse tree in the window of (-3, 3) words around target word.
- Collocations formed with the parents of the surrounding words in the window (-3, 3) words around the target word (COL_PARENT).
- Occurrences in the current sentence of the words that are linked to the current word with a semantic relation of AGENT or THEME in WordNet 2.0 glosses (XWN_LEMMA).
  We used files from XWN project (XWN, 2003) containing WordNet 2.0 glosses that were sense disambiguated and tagged with semantic relations both manually and automatically. For each word to be disambiguated we created a signature consisting of the set of words that are linked with a semantic relation of THEME or AGENT in all WordNet glosses. For every word in this set we created a feature showing if that word appears in the current sentence containing the target word.

Then we added a new feature consisting of all the named entities in a window of (-5,5) sentences around the target word. We called this feature NAMED_ENTITIES. We created the feature set FS_3 by adding this new feature to FS_2.

In the end we applied a greedy feature selection algorithm to features in FS_3 inspired by (Mihalcea, 2002). Because feature selection was running very slow, the feature selection algorithm was run

CTX_WORD_1	CTX_WORD_-2	CTX_LEMMA_1	COL_POS_-2_0
CTX_POS_1	CTX_WORD_-1	CTX_LEMMA_2	COL_LEMMA_0_1
CTX_WORD_2	COL_PARENT_-3_-1	CTX_LEMMA_3	COL_PARENT_-2_2
CRT_WORD	COL_PARENT_-3_2	NAMED_ENTITIES	CTX_POS_3
CTX_WORD_-3	CTX_WORD_3	COL_PARENT_-1_1	COL_WORD_-1_1

Table 1: The feature set FS_4 obtained from the features most selected by the greedy selection algorithm applied to all the words in Senseval 2

only for words in Senseval 2 English lexical sample task and the top 20 features appearing the most often (at least 5 times) in the selected feature set for each word were used to create feature set FS_4 presented in table 1.

Corpus	Precision
SemCor	79.61%
XWN	57.21%
SemCor+XWN	79.44%

Table 3: The precision using SemCor and disambiguated glosses from XWN project

3 Experiments and results

For SemEval 2007 we performed several experiments: we tested ME and SVM classifiers on the 4 feature sets described in the previous section and then we tried to improve the performance using disambiguated glosses from XWN project. Each set of experiments together with the final submission is described in detail below.

3.1 Experiments with different feature sets

Initially we made experiments with the set of features used at Senseval 3 All Words task. For training the ME and SVM classifiers, we used a combined corpus made from SemCor, Senseval 3 All Words corpus, Senseval 3 Lexical Sample testing and training corpora and Senseval 2 Lexical sample training corpus. For testing we used Senseval 2 Lexical Sample corpus. We made 3 experiments for the first three feature sets FS_1, FS_2, FS_3. Both algorithms attempted to disambiguate all the words (coverage=100%) so the precision is equal with recall. The precision of each algorithm on each feature set is presented in table 2.

Algorithm	FS_1	FS_2	FS_3	FS_4
ME	76.03%	75.86%	76.03%	77.56%
SVM	73.30%	71.36%	71.46%	71.90%

Table 2: The precision of ME and SVM classifiers using 4 sets of features.

After the first 3 experiments we noticed that both ME and SVM classifiers had good results using the first set of features FS_1. This seemed odd since we expected an increase in performance with the additional features. This led us to the idea that not all the features are useful for all words. So we created a greedy feature selection algorithm based on the performance of the SVM classifier (Mihalcea, 2002). The feature selection algorithm starts with an empty set of features S, and iteratively adds one feature from the set of unused features U. Initially the set U contains all the features. The algorithm iterates as long as the overall performance increase. At each step the algorithm adds tentatively one feature from the set U to the existing feature list S and measures the performance of the classifier on a 10 fold cross validation on the training corpus. The feature providing the greatest increase in performance is finally added to S and removed from U.

The feature selection algorithm turned out to be very slow, so we could not use it to train all the words. Therefore we used it to train only the words from Senseval 2 Lexical Sample task and then we computed a global set of features by selecting the first 20 features that were selected the most (at least 5 times).

This list of features was named FS_4. Table 2 that SVM classifier with FS_4 did not get a better performance than FS_1 while ME surprisingly did get 1.53% increase in performance. Given the higher precision of ME classifier, it was selected for creating the submission file.

3.2 Experiments using disambiguated glosses from XWN project

The ME classifier works well for words with enough training examples. However we found many words for which the number of training examples was too small. We tried to increase the number of training examples using the disambiguated WordNet glosses from XWN project. Not all the senses in the disambiguated glosses were assigned manually and the text of the glosses is different than normal running text. However we were curious if we could improve the overall performance by adding more training examples. We made 3 experiments showed in table 3. For all three experiments we used Senseval 2 English All Words corpus for testing. On the first experiment we used SemCor for training, on the second we used disambiguated glosses from XWN project and on the third we used both. XWN did not bring an improvement to the overall precision, so we decided to use XWN as a fall back strategy only for 70 words that did not have training examples is other corpora.

3.3 Final Submission

For final submission we used trained ME models using feature set FS_4 for 852 words, representing 1715 instances using SemCor, Senseval 2 and 3 English All Words and Lexical Sample testing and training and OMWE 1.0. For 50 words representing 70 instances, we used disambiguated WordNet glosses from XWN project to train ME classifiers using feature set FS_4. For the rest of 484 words for which we could not find training examples we used the First Sense in WordNet strategy. The submitted answer had a 100% coverage and a 81.446% precision presented in table 4.

LCC-WSD	81.446%
Best submission	83.208%

Table 4: The LCC-WSD and the best submission at SemEval 2007 Coarse All Words Task

4 Conclusions

LCC-WSD team used two supervised approaches for performing experiments using coarse grained senses: Maximum Entropy and Support Vector Machines. We used 4 feature sets: the first one was the feature set used in Senseval 3 and next two representing incremental additions. The fourth feature set represents a global set of features obtained from the individual feature sets for each word resulted from the greedy feature selection algorithm used to improve the performance of SVM classifiers. In addition we used disambiguated WordNet glosses from XWN to measure the improvement made by adding additional training examples. The submitted answer has a coverage of 100% and a precision of 81.446%.

References

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Elliot Glaysher and Dan I. Moldovan. 2006. Speeding up full syntactic parsing by leveraging partial parsing decisions. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 295–300, Sydney, Australia. Association for Computational Linguistics.

Zhang Le, 2006. *Maximum Entropy Modeling Toolkit for Python and C++*. Software available at http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.

Rada Mihalcea. 2002. Instance based learning with automatic feature selection applied to word sense disambiguation. In *Proceedings of the 19th International Conference on Computational Linguistics COLING 2002*, Taiwan.

Hwee Tou Ng and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 40–47, Morristown, NJ, USA. Association for Computational Linguistics.

Mark Sanderson. 1994. Word sense disambiguation and information retrieval. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 49–57, Dublin, IE.

XWN, 2003. *eXtended WordNet*. Software available at http://xwn.hlt.utdallas.edu.