SenseLearner: Word Sense Disambiguation for All Words in Unrestricted Text

Rada Mihalcea and Andras Csomai
Department of Computer Science and Engineering
University of North Texas
rada@cs.unt.edu, ac0225@unt.edu

Abstract

This paper describes SENSELEARNER – a minimally supervised word sense disambiguation system that attempts to disambiguate all content words in a text using WordNet senses. We evaluate the accuracy of SENSELEARNER on several standard sense-annotated data sets, and show that it compares favorably with the best results reported during the recent SENSEVAL evaluations.

1 Introduction

The task of word sense disambiguation consists of assigning the most appropriate meaning to a polysemous word within a given context. Applications such as machine translation, knowledge acquisition, common sense reasoning, and others, require knowledge about word meanings, and word sense disambiguation is considered essential for all these applications.

Most of the efforts in solving this problem were concentrated so far toward targeted supervised learning, where each sense tagged occurrence of a particular word is transformed into a feature vector, which is then used in an automatic learning process. The applicability of such supervised algorithms is however limited only to those few words for which sense tagged data is available, and their accuracy is strongly connected to the amount of labeled data available at hand.

Instead, methods that address all words in unrestricted text have received significantly less attention. While the performance of such methods is usually exceeded by their supervised lexical-sample alternatives, they have however the advantage of providing larger coverage.

In this paper, we present a method for solving the semantic ambiguity of all content words in a text. The algorithm can be thought of as a minimally supervised word sense disambiguation algorithm, in that it uses a relatively small data set for training purposes, and generalizes the concepts learned from the training data to disambiguate the words in the test data set. As a result, the algorithm does not need a separate classifier for each word to be disambiguated, but instead it learns global models for general word categories.

2 Background

For some natural language processing tasks, such as part of speech tagging or named entity recognition, regardless of the approach considered, there is a consensus on what makes a successful algorithm. Instead, no such consensus has been reached yet for the task of word sense disambiguation, and previous work has considered a range of knowledge sources, such as local collocational clues, common membership in semantically or topically related word classes, semantic density, and others.

In recent SENSEVAL-3 evaluations, the most successful approaches for all words word sense disambiguation relied on information drawn from annotated corpora. The system developed by (Decadt et al., 2004) uses two cascaded memory-based classifiers, combined with the use of a genetic algorithm for joint parameter optimization and feature selection. A separate "word expert" is learned for each ambiguous word, using a concatenated corpus of English sense-

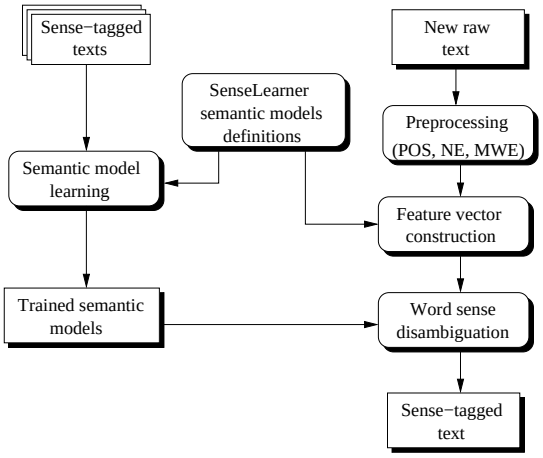

Figure 1: Semantic model learning in SENSELEARNER

tagged texts, including SemCor, SENSEVAL data sets, and a corpus built from WordNet examples. The performance of this system on the SENSEVAL-3 English all words data set was evaluated at 65.2%.

Another top ranked system is the one developed by (Yuret, 2004), which combines two Naive Bayes statistical models, one based on surrounding collocations and another one based on a bag of words around the target word. The statistical models are built based on SemCor and WordNet, for an overall disambiguation accuracy of 64.1%.

A different version of our own SENSELEARNER system (Mihalcea and Faruque, 2004), using three of the semantic models described in this paper, combined with semantic generalizations based on syntactic dependencies, achieved a performance of 64.6%.

3 SenseLearner

Our goal is to use as little annotated data as possible, and at the same time make the algorithm *general* enough to be able to disambiguate as many content words as possible in a text, and *efficient* enough so that large amounts of text can be annotated in real time. SENSELEARNER is attempting to learn general semantic models for various word categories, starting with a relatively small sense-annotated corpus. We base our experiments on SemCor (Miller et al., 1993), a balanced, semantically annotated dataset, with all content words manually tagged by trained lexicographers.

The input to the disambiguation algorithm consists of raw text. The output is a text with word meaning annotations for all open-class words.

The algorithm starts with a preprocessing stage, where the text is tokenized and annotated with part-of-speech tags; collocations are identified using a sliding window approach, where a collocation is defined as a sequence of words that forms a compound concept defined in WordNet (Miller, 1995); named entities are also identified at this stage[1].

Next, a semantic model is learned for all predefined word categories, which are defined as groups of words that share some common syntactic or semantic properties. Word categories can be of various granularities. For instance, using the SENSELEARNER learning mechanism, a model can be defined and trained to handle all the *nouns* in the test corpus. Similarly, using the same mechanism, a finer-grained model can be defined to handle all the verbs for which at least one of the meanings is of type *<move>*. Finally, small coverage models that address one word at a time, for example a model for the adjective *small*, can be also defined within the same framework. Once defined and trained, the models are used to annotate the ambiguous words in the test corpus with their corresponding meaning. Section 4 below provides details on the various models that are currently implemented in SENSELEARNER, and information on how new models can be added to the SENSELEARNER framework.

Note that the semantic models are applicable only to: (1) words that are covered by the word category defined in the models; and (2) words that appeared at least once in the training corpus. The words that are not covered by these models (typically about 10-15% of the words in the test corpus) are assigned with the most frequent sense in WordNet.

An alternative solution to this second step was suggested in (Mihalcea and Faruque, 2004), using semantic generalizations learned from dependencies identified between nodes in a conceptual network. Their approach however, although slightly more accurate, conflicted with our goal of creating an *efficient* WSD system, and therefore we opted for the simpler backoff method that employs WordNet sense frequencies.

[1]We only identify *persons*, *locations*, and *groups*, which are the named entities specifically identified in SemCor.

4 Semantic Models

Different semantic models can be defined and trained for the disambiguation of different word categories. Although more general than models that are built individually for each word in a test corpus (Decadt et al., 2004), the applicability of the semantic models built as part of SENSELEARNER is still limited to those words previously seen in the training corpus, and therefore their overall coverage is not 100%.

Starting with an annotated corpus consisting of all annotated files in SemCor, a separate training data set is built for each model. There are seven models provided with the current SENSELEARNER distribution, implementing the following features:

4.1 Noun Models

modelNN1: A contextual model that relies on the first noun, verb, or adjective before the target noun, and their corresponding part-of-speech tags.
modelNNColl: A collocation model that implements collocation-like features based on the first word to the left and the first word to the right of the target noun.

4.2 Verb Models

modelVB1 A contextual model that relies on the first word before and the first word after the target verb, and their part-of-speech tags.
modelVBColl A collocation model that implements collocation-like features based on the first word to the left and the first word to the right of the target verb.

4.3 Adjective Models

modelJJ1 A contextual model that relies on the first noun after the target adjective.
modelJJ2 A contextual model that relies on the first word before and the first word after the target adjective, and their part-of-speech tags.
modelJJColl A collocation model that implements collocation-like features using the first word to the left and the first word to the right of the target adjective.

4.4 Defining New Models

New models can be easily defined and trained following the same SENSELEARNER learning methodology. In fact, the current distribution of SENSELEARNER includes a template for the subroutine required to define a new semantic model, which can be easily adapted to handle new word categories.

4.5 Applying Semantic Models

In the training stage, a feature vector is constructed for each sense-annotated word covered by a semantic model. The features are model-specific, and feature vectors are added to the training set pertaining to the corresponding model. The label of each such feature vector consists of the target word and the corresponding sense, represented as *word#sense*. Table 1 shows the number of feature vectors constructed in this learning stage for each semantic model.

To annotate new text, similar vectors are created for all content-words in the raw text. Similar to the training stage, feature vectors are created and stored separately for each semantic model.

Next, word sense predictions are made for all test examples, with a separate learning process run for each semantic model. For learning, we are using the Timbl memory based learning algorithm (Daelemans et al., 2001), which was previously found useful for the task of word sense disambiguation (Hoste et al., 2002), (Mihalcea, 2002).

Following the learning stage, each vector in the test data set is labeled with a *predicted* word and sense. If several models are simultaneously used for a given test instance, then all models have to agree in the label assigned, for a prediction to be made. If the word predicted by the learning algorithm coincides with the target word in the test feature vector, then the predicted sense is used to annotate the test instance. Otherwise, if the predicted word is different than the target word, no annotation is produced, and the word is left for annotation in a later stage.

5 Evaluation

The SENSELEARNER system was evaluated on the SENSEVAL-2 and SENSEVAL-3 English all words data sets, each data set consisting of three texts from the Penn Treebank corpus annotated with WordNet senses. The SENSEVAL-2 corpus includes a total of 2,473 annotated content words, and the SENSEVAL-3 corpus includes annotations for an additional set of 2,081 words. Table 1 shows precision and recall figures obtained with each semantic model on these two data sets. A baseline, computed using the most frequent sense in WordNet, is also indicated. The best results reported on these data sets are 69.0% on SENSEVAL-2 data (Mihalcea and Moldovan, 2002),

Model	Training size	SENSEVAL-2		SENSEVAL-3	
		Precision	Recall	Precision	Recall
modelNN1	88058	0.6910	0.3257	0.6624	0.3027
modelNNColl	88058	0.7130	0.3360	0.6813	0.3113
modelVB1	48328	0.4629	0.1037	0.5352	0.1931
modelVBColl	48328	0.4685	0.1049	0.5472	0.1975
modelJJ1	35664	0.6525	0.1215	0.6648	0.1162
modelJJ2	35664	0.6503	0.1211	0.6593	0.1153
modelJJColl	35664	0.6792	0.1265	0.6703	0.1172
model*1/2	207714	0.6481	0.6481	0.6184	0.6184
model*Coll	172050	0.6622	0.6622	0. 6328	0.6328
Baseline		63.8%	63.8%	60.9%	60.9%

Table 1: Precision and recall for the SENSELEARNER semantic models, measured on the SENSEVAL-2 and SENSEVAL-3 English all words data. Results for combinations of contextual (model*1/2) and collocational (model*Coll) models are also included.

and 65.2% on SENSEVAL-3 data (Decadt et al., 2004). Note however that both these systems rely on significantly larger training data sets, and thus the results are not directly comparable.

In addition, we also ran an experiment where a separate model was created for each individual word in the test data, with a back-off method using the most frequent sense in WordNet when no training examples were found in SEMCOR. This resulted into significantly higher complexity, with a very large number of models (about 900–1000 models for each of the SENSEVAL-2 and SENSEVAL-3 data sets), while the performance did not exceed the one obtained with the more general semantic models.

The average disambiguation precision obtained with SENSELEARNER improves significantly over the simple but competitive baseline that selects by default the "most frequent sense" from WordNet. Not surprisingly, the verbs seem to be the most difficult word class, which is most likely explained by the large number of senses defined in WordNet for this part of speech.

6 Conclusion

In this paper, we described and evaluated an efficient algorithm for minimally supervised word-sense disambiguation that attempts to disambiguate all content words in a text using WordNet senses. The results obtained on both SENSEVAL-2 and SENSEVAL-3 data sets are found to significantly improve over the simple but competitive baseline that chooses by default the most frequent sense, and are proved competitive with the best published results on the same data sets. SENSELEARNER is publicly available for download at *http://lit.csci.unt.edu/˜senselearner*.

Acknowledgments

This work was partially supported by a National Science Foundation grant IIS-0336793.

References

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.

B. Decadt, V. Hoste, W. Daelemans, and A. Van den Bosch. 2004. Gambl, genetic algorithm optimization of memory-based wsd. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, July.

V. Hoste, W. Daelemans, I. Hendrickx, and A. van den Bosch. 2002. Evaluating the results of a memory-based word-expert approach to unrestricted word sense disambiguation. In *Proceedings of the ACL Workshop on "Word Sense Disambiguatuion: Recent Successes and Future Directions"*, Philadelphia, July.

R. Mihalcea and E. Faruque. 2004. SenseLearner: Minimally supervised word sense disambiguation for all words in open text. In *Proceedings of ACL/SIGLEX Senseval-3*, Barcelona, Spain, July.

R. Mihalcea and D. Moldovan. 2002. Pattern learning and active feature selection for word sense disambiguation. In *Senseval 2001, ACL Workshop*, pages 127–130, Toulouse, France, July.

R. 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)*, Taipei, Taiwan, August.

G. Miller, C. Leacock, T. Randee, and R. Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, New Jersey.

G. Miller. 1995. Wordnet: A lexical database. *Communication of the ACM*, 38(11):39–41.

D. Yuret. 2004. Some experiments with a naive bayes wsd system. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, July.