

## LCC's WSD Systems for Senseval 3

Adrian Novischi, Dan Moldovan, Paul Parker, Adriana Badulescu, Bob Hauser

Language Computer Corporation

1701 N Collins Blvd.

Richardson, TX 75080

{adrian,moldovan,parker,adriana,hauser}@languagecomputer.com

### Abstract

This document provides a description of the Language Computer Corporation (LCC) Systems that participated in the Senseval 3 English All Words and Semantic disambiguation of WordNet glosses task. Both systems consist of a set of disambiguation methods. English All Words system includes methods based on heuristics that exploit external resources, methods using the senses of the previous disambiguated words and methods based on supervised machine learning algorithms. The WSD system for Semantic Disambiguation of WordNet glosses contains methods based on heuristics exploiting WordNet lexical database, the Semcor Corpus and domain labels assigned to noun synsets. All these methods are combined using rules that optimize their output.

### 1 Introduction

Polysemy, or lexical ambiguity is the property of some words to have multiple meanings or senses. The definition and the number of senses is a function of the dictionary used. In the Senseval 3 English All Words task the dictionary chosen was WordNet 1.7.1 (Miller, 1995). In the task of Semantic Disambiguation of WordNet glosses, the dictionary chosen was WordNet 2.0. The Word Sense Disambiguation (WSD) problem is the labeling of each content word with the most appropriate sense. Both LCC WSD Systems that participated in Senseval 3 contain a set of disambiguation methods. In the case of the system for English All Words task, some of the methods are based on heuristics, some are based on machine learning algorithms, and others make use of previous disambiguated words. In the case of WSD system for Semantic Disambiguation of WordNet glosses, the disambiguation methods consist of heuristics based on WordNet lexical database, SemCor corpus (Miller et al., 1994), and domain labels assigned to noun synsets. In the end the

methods are combined using rules (Magnini and Cavaglia, 2000). Before effective disambiguation takes place some preprocessing of the input text is required. Preprocessing consists of token identification, part of speech tagging (Brill, 1995), Named Entity recognition (Surdeanu and Harabagiu, 2002) and parsing (Collins, 1997).

Section 2 explains the disambiguation methods used by the system. Section 3 describes the approach for combining these disambiguation methods. Section 4 discusses the results and the final section draws the conclusions.

### 2 Methods description

Each of the two WSD system is based on different sets of Disambiguation methods.

#### 2.1 Semantic Disambiguation of WordNet glosses

The heuristics used for semantic disambiguation of WordNet glosses were first described in (Harabagiu et al., 1999), (Mihalcea and Moldovan, 2001a), (Novischi, 2002), (Novischi, 2004). This paper provides only a short description of each method:

**Same hierarchy relation** - assigns a noun or verb the sense that is an ancestor of the synset of the gloss.

**Lexical parallelism** - identifies the words with the same part of speech separated by commas or conjunctions and marks them with the senses that belong to the same hierarchy (for nouns and verbs) or to the same cluster (for adjectives). In the case of nouns and verbs this method can find multiple pairs of senses belonging to different hierarchies.

**SemCor previous word** - given a word in a gloss this heuristic forms a pair with the previous word in a gloss and searches for this pair in the SemCor corpus. If in all occurrences of this pair the given word has the same sense, then the heuristic assigns this sense to the target word.

**SemCor next word** - the same as the previous heuristic but this heuristic forms a pair from the target word in a gloss and the next word and searches for this pair in the SemCor corpus. If the target word has the same sense in all occurrences of the pair, then the heuristic assigns this sense to the target word.

**Cross reference** - given an ambiguous word  $W$  in the synset  $S$ , this method looks for a reference to the synset  $S$  in all the glosses corresponding to word  $W$ 's senses. By reference to a word  $W$  we denote a word or a part of a compound concept that has the same lemma as the word  $W$ .

**Reversed cross reference** - given a word  $W$  in the gloss  $G$  of the synset  $S$ , this method assigns to the word  $W$  the sense that contains in its set of synonyms one of the words from the gloss  $G$ .

**Distance among glosses** - determines the number of common word lemmas between two glosses. For an ambiguous word  $W$  in a gloss  $G$  this method selects the sense with the greatest number of common word lemmas with the gloss  $G$ .

**Common Domain** - using the domain labels assigned by (Magnini and Cavaglia, 2000) this method selects the sense of a word in a gloss that has the same domain as the synset of the gloss.

**Patterns** - exploits the idiosyncratic nature of the WordNet glosses. The patterns of the form "N successive words" and "M words ... N words" are extracted offline from the WordNet glosses and manually disambiguated. This method matches the patterns against glosses and assigns to the words the corresponding sense in the pattern.

**First Sense Restricted** - this method assigns sense 1 to a noun or verb if this sense has the smallest number of ancestors in the ISA hierarchy from all senses (it is the most general sense). The method selects sense 1 for an adjective if this sense has the greatest number of similarity pointers compared to all the other senses.

## 2.2 Disambiguation methods for English All Words task

The disambiguation methods in this task can be classified in three types described below: methods based on heuristics, machine learning disambiguation methods, and incremental disambiguation methods.

### 2.2.1 WSD Methods based on Heuristics

In this category the methods can be further classified in methods that use hand coded rules, methods that use WordNet and methods that use SemCor (Miller et al., 1994). Some methods applied to Semantic Disambiguation of WordNet glosses were also applied on open text: Semcor Previous Word, Semcor Next Word, Patterns and Lexical Parallelism.

### 2.2.2 Machine Learning Disambiguation methods

We used Support Vector Machines (Cortes and Vapnik, 1995) for disambiguating verbs, adjectives, and adverbs, and C5.0 (Quinlan, 2003) rules for disambiguating nouns.

#### Support Vector Machines (SVM) Method

We used the following set of features (Mihalcea, 2002): current word form and part of speech, contextual features, collocations in a window of (-3,3) words, and keywords and bigrams in a window of (-3,3) sentences. For disambiguating verbs we used an additional set of features: Verb mode (which can take 4 values: ACTIVE, INFINITIVE, PAST, GERUND), verb voice (which can take 2 values ACTIVE, PASSIVE), the parent of the current verb in the parse tree (ex: VP, NP), The first ancestor that is not VP in the parse tree (like S, NP, PP, SBAR).

We generated feature values using the Senseval 2 Lexical Sample training corpus and SemCor corpus combined, and we trained the SVM classifier on those words that had at least 10 training examples.

#### C5.0 Rules Method

The second machine learning method, used for word sense disambiguating nouns, is the C5.0 decision tree learning (Quinlan, 2003). For generating training examples we used the features described in (Mihalcea and Moldovan, 2001b). We used the SemCor 1.7.1 collection and Senseval 2 All Words collection as training corpus.

### 2.2.3 Incremental Disambiguation

The main idea of incremental disambiguation is to disambiguate new words using senses of the previous disambiguated words. We used some of the procedures presented in (Mihalcea and Moldovan, 2001b) briefly described below:

**WordNet distance 0 with disambiguated words** - find words that are in the

same synset with already disambiguated words.

**WordNet distance 1 with disambiguated words** - find words that are in a hypernymy/hyponymy relation with the words already disambiguated.

### 3 Combining Methods using Rules

At this point each WSD system has a pool of disambiguation methods. An approach using rules for selecting the right sense was described in (Novischi, 2004) and is summarized below. For a given disambiguated word we create a training example for each of its senses. Given a manually-disambiguated corpus, training examples are generated for each sense of the tagged words with their correct classification. Then we can train a machine learning algorithm on this set of training examples. If we use C4.5 or C5.0, where each rule has an associated accuracy value, we can output the single sense that is classified as CORRECT by the rule with the best accuracy. For Semantic disambiguation of WordNet glosses task, we used a set of rules given by C4.5 program trained on a set of training examples generated from 3196 goldstandard glosses. For the English All Words task, however, we created the rules manually, each rule corresponding to a disambiguation method and assigned an accuracy value equal to the precision of the disambiguation method. The rules were tested in the decreasing order of their accuracy.

## 4 Results

We measured the accuracy of the LCC WSD systems using precision, recall and coverage. Precision is defined as the number of correct disambiguated words over the number of attempted words. Recall is defined as the number of correct disambiguated words over the total number of words for disambiguation. Coverage is defined as the number of attempted words over the total number of words.

### 4.1 English All Words

In order to illustrate the performance of the system, we examine it from three different perspectives. First we ran the system without defaulting to first sense and computed the results excluding monosemous words (row a in table 1). Second we ran the system including defaulting to sense 1, but we still excluded monosemous words when we computed the results (row b in table 1). Third we included both the first

sense for words that were not otherwise disambiguated by the system and monosemous words (row c in table 1). Table 1 presents the results given by the Python scorer script. Precision and recall in rows a and b do not include the monosemous words.

### 4.2 Semantic Disambiguation of WordNet glosses

In order to illustrate the performance of the system, we examined it from two different perspectives. First we ran the system including assigning sense 1 for words that were not otherwise disambiguated by the system (row a. in table 2). Second we ran the system without assigning sense 1 to words that were not otherwise disambiguated by the system (row b. in table 2).

## 5 Conclusion

The LCC WSD systems for Senseval 3 employ sets of disambiguation methods. Some methods are based on hand coded rules, some use information in WordNet, and some exploit the statistics of the SemCor corpus. Others use the senses of previous disambiguated words and still others are based on supervised machine learning algorithms. All these methods are combined using rules that optimize their output. The way the WSD systems are designed using multiple methods for disambiguation ensures a greater coverage than a single method by itself. Combining methods using machine learning algorithms improves the precision of the system.

## References

- Eric Brill. 1995. Unsupervised learning of disambiguation rules for part of speech tagging. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 1–13, Somerset, New Jersey. Association for Computational Linguistics.
- Michael Collins. 1997. Three generative lexicalized models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL 1997)*.
- Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.
- S. Harabagiu, G. Miller, and D. Moldovan. 1999. WordNet 2 - a morphologically and semantically enhanced resource. In *Proceed-*

	Experiment	Precision	Recall	Coverage
a	LCC WSD alone	53.6%	41.4%	77.3%
b	LCC WSD + First Sense	53.9%	54.9%	101.9%
c	LCC WSD + First Sense + Monosemous	60.7%	60.7%	100.0%

Table 1: Performance of the LCC WSD system in English All Words Task

	Experiment	Precision	Recall	Coverage
a	LCC WSD	65.1%	62.2%	95.5%
b	LCC WSD without First Sense	70.1%	50.4%	71.9%

Table 2: Performance of the LCC WSD system in WordNet glosses task

- ings of SIGLEX-99*, pages 1–8, University of Maryland.
- B. Magnini and G. Cavaglia. 2000. Integrating subject field codes into WordNet. In *Proceedings of the LREC-2000*, Athens, Greece.
- Rada Mihalcea and Dan Moldovan. 2001a. eXtended WordNet: progress report. In *Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*, pages 95–100, Pittsburgh, PA.
- Rada Mihalcea and Dan I. Moldovan. 2001b. A highly accurate bootstrapping algorithm for word sense disambiguation. *International Journal on Artificial Intelligence Tools*, 10(1-2):5–21.
- 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.
- M. Miller, G. and Chodorow, S. Landes, C. Leacock, and R. Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 240–243.
- G. Miller. 1995. Wordnet: a lexical database. *Communications of the ACM*, 38(11):39–41.
- Adrian Novischi. 2002. Accurate semantic annotations via pattern matching. In *Proceedings of Florida Artificial Intelligence Research Society*, pages 375–379, Pensacola Beach, Florida, USA.
- Adrian Novischi. 2004. Combining methods for word sense disambiguation of WordNet glosses. In *Proceedings of Florida Artificial Intelligence Research Society*, Miami Beach, Florida, USA.
- Ross Quinlan. 2003. Data Mining Tools See5 and C5.0., <http://www.rulequest.com/see5-info.html>.
- Mihai Surdeanu and Sanda Harabagiu. 2002. Infrastructure for open-domain information extraction. In *Proceedings of the Human Language Technology Conference*, San Diego, California.