Combining Methods for Word Sense Disambiguation of WordNet Glosses

Adrian Novischi
Language Computer Corporation
1701 N. Collins Blvd. Suite 2200
Richardson, Texas, 75080
adrian@languagecomputer.com

Abstract

This paper presents a new approach for combining different semantic disambiguation methods that are part of a Word Sense Disambiguation(WSD) system. The way these methods are combined greatly influences the overall system performance. The approach is based on generating training examples, for each sense of the word, based on the output of each disambiguation method. A set of rules is learned from the training examples and then applied to optimize the output of the WSD system. We tested this approach on disambiguating WordNet glosses. However the approach is applicable to any WSD system. Our approach yielded a 3% gain in performance when compared with more traditional approaches such as selecting the sense given by the best disambiguation method or summing up the contribution of each method.

Introduction

Word Sense Disambiguation is a challenging problem. Assigning the correct meaning to a polysemous word in a context, requires morphologic, syntactic, semantic and pragmatic knowledge sources. The criteria for selecting the right sense of a word depends on the word itself, its part of speech and the context in which it appears. From this point of view a suitable approach for word sense disambiguation is the word expert approach, in which for each word, a WSD system learns knowledge that will help to assign the correct sense to that word in future contexts (Yarowsky 2000), (Mihalcea 2002). However, the training data, consisting of contexts in which the word appears with its correct sense, is often insufficient for learning. One possible solution is to group words with similar properties into classes and then disambiguate classes of words instead of a single word. In this case the set of contexts in which a class of words can appear is much larger than the set of contexts for a single word. For example Yarowsky, (1992), groups the words listed in the same category of Roget's International Thesaurus, and therefore avoids the knowledge acquisition bottleneck.

In WSD a word can be assigned to more than one class and can be disambiguated by several methods. For a specific instance of WSD some methods can assign the correct sense, others the wrong one. The problem is how to choose the correct sense from the output given by several methods. In this paper we will overview several strategies for combining semantic disambiguation methods: the sense given by the best method, summing up the classification of each method and machine learning techniques like decision lists and C4.5rules (Quinlan 1993). Although the strategies described are applicable for any semantic disambiguation system, we used them for the disambiguation of WordNet glosses.

The idea of combining different knowledge sources is not new and is used in almost any approach to word sense disambiguation. Combining the results of different methods of word sense disambiguation is a particular case of this general idea. Rigau, Atserias, & Agirre, (1997) used several heuristics for semantic disambiguation and they combined them by summing up the weights assigned by each heuristic to each sense. The word sense disambiguation approach proposed by Mihalcea & Moldovan, (2000) used a set of procedures that are called in a given order and each subsequent procedure attempts to disambiguate a word if the previous procedures failed to do so. On the SENSEVAL-2 competition the WSD system from Stanford University was based on a combination of 23 WSD classifiers made by the students taking the Natural Language Processing course (Klein *et al.* 2002). Their system had three voting schemes: majority voting, weighted voting and maximum entropy model.

In this paper we propose a new approach for combining semantic disambiguation methods based on rules extracted from training examples. Each sense of a word in a given context can be right or wrong. Each method can assign a label to a sense: correct, incorrect or unknown. By combining the labels given by each method with the correct classification of a given sense of a word in a context, we generate training examples from which we extract rules for selecting the right sense. For extracting the rules we used decision lists and C4.5 rules. We compared this approach with selecting the sense given by the most accurate method and summing the contribution of each method to a given sense. Section 2 ("Semantic Disambiguation of WordNet glosses") delineates the need of disambiguating WordNet glosses and the resources used for disambiguation. Section 3 ("Combining Semantic Disambiguation Methods") describes each method used for semantic disambiguation and the method combining strategies. Section 4 ("Example") provides an example, and section 5 ("Results") presents the results. The final sec-

tion draws the conclusions and discusses further work.

Semantic Disambiguation of WordNet glosses

The semantic disambiguation of WordNet glosses is part of eXtended WordNet project (http://xwn.hlt.utdallas.edu) whose target is to extract new information from WordNet glosses (Miller 1995) and to overcome the limited number of connections between topically related words. By linking each word in a gloss to each corresponding concept, we can increase the number of connections between topically related concepts: by adding a link between two concepts that appear in the same gloss or between these concepts and the synset of the gloss.

Semantic disambiguation of WordNet glosses is different from semantic disambiguation of open text. First, the meaning of the concept to which the gloss belongs restricts for the possible senses of the open class words in the gloss. Second, the gloss is not a complete sentence. Several transformations have to be made in order to transform a gloss into a sentence. Third, the glosses have an idiosyncratic nature containing patterns in which the open class words have the same sense. Fourth, we cannot train statical methods on a set of disambiguated glosses, since a lot of words appears few times in WordNet glosses and do not exhibit the entire range of senses. A set of semantic disambiguation glosses is more useful for testing than for training. For semantic disambiguation of WordNet glosses, we rely on a set of heuristics that exploit the information contained in WordNet or are based on external resources like SemCor. SemCor (Miller *et al.* 1994) is a corpus formed with about 25% of the Brown Corpus files having all the words part of speech tagged and semantically disambiguated. The SemCor corpus was annotated with senses from WordNet 1.6. In order to disambiguate using WordNet 2.0 we automatically transformed the senses from WordNet 1.6 to WordNet 2.0 using the sense keys.

Combining Semantic Disambiguation Methods

The heuristics used for semantic disambiguation of WordNet glosses were first described in (Harabagiu, Miller, & Moldovan 1999), (Mihalcea & Moldovan 2001) and (Novischi 2002). In this paper I will provide only a short description for each method:

Same hierarchy relation - assigns to a noun or verb in a gloss 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 belongs 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 pair 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 this pair in 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 this pair in 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 understand a word or a part of 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 - Magnini & Cavaglia (2000) assigned a domain label to all the noun synsets in WordNet 1.6. We translated these domains for the noun synsets in WordNet 2.0. Using these synset labels, 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 are selected and manually disambiguated. This method matches the patterns against glosses and assign to the words the corresponding sense in the pattern.

First Sense Restricted - this method assign 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 than all the other senses. This methods is based on the intuition that, in WordNet glosses, most of the words are used in their most general sense.

Combining Methods

Each of the method tags a word with a sense. Between two methods that tag a word with different senses, we would like to know which one is right and which one is wrong. If a method assigns a sense to a given word we can consider that the method assigns the tag CORRECT to that sense and INCORRECT to the other senses. If the method does not assign a sense to that word, we can consider that the method assigns the tag UNKNOWN to all senses. The final classification of a sense of a word is CORRECT or INCORRECT.

For a disambiguated word taken from gold-standard, we can generate a number of training examples equal to the number of senses, each training example corresponding to a sense. Each training example contains as attributes the tags assigned by each method to the corresponding sense. In the training example we also include the part of speech of the word, e.g.: NOUN, VERB, ADJ, ADV. We can also add as attributes the output returned by other features. For example we can consider as a feature the fact that the given sense corresponding to the training example is the hypernym

of the synset of the gloss. This feature can output TRUE or FALSE. In the current implementation we do not have any attributes other than the tags assigned by the disambiguated methods and the part of speech tagger.

For each gloss in the gold standard file and for each open class word in the gloss we generate a set of training examples. From all training example we extract rules using different machine learning techniques. Then, we use the learned rules to optimize the output of a WSD system that is based on several disambiguation methods. For each sense of a new word, given the output of the disambiguation methods, we select a label for that sense: CORRECT or INCORRECT. To each sense we assign a value given by the accuracy of the rule. We assign a positive number to a sense labeled as CORRECT and a negative number given by the negated accuracy value of the rule if the sense is labeled as INCORRECT. Then we select as correct the sense with the highest score.

An advantage of this approach is the ability to incorporate incomplete sources of information as attributes. As previously stated, we add attributes that output TRUE or FALSE. We also have methods that say only when one sense is INCORRECT. Sometimes in a given context for a word, we cannot easily explain why a given sense is correct, but we can very easily explain (and implement as a computer procedure) why a sense is incorrect. Another advantage is that this approach allows a WSD system to be easily improved. Analyzing the output of the system one can observe the conditions when a given method mistakes the sense of a word. Then, those conditions can be implemented as a feature in the program, that outputs TRUE or FALSE, and the system automatically generates a rule that states the fact that the method gives incorrect results when the feature is TRUE.

Example

Using as features all the methods described in the previous section and the part of speech of the word, we attempt to disambiguate the noun *money* in the gloss of sense 3 of *finance* in WordNet 2.0:

finance#3 – the management of money and credit and banking and investments

The noun *money* has three senses:

money#1 – the most common medium of exchange; functions as legal tender; "we tried to collect the money he owed us"

money#2 – wealth reckoned in terms of money; "all his money is in real estate"

money#3 – the official currency issued by a government or national bank; "he changed his money into francs"

Applying the disambiguation methods described in the previous section we achieved the following results:

The **Patterns** and **SemCor previous** methods assigned sense 1.

The **Domains** method assigned sense 2.

The **Gloss distance** method assigned sense 3.

The **Lexical parallelism** method labeled all three senses as correct.

The correct sense of the noun money, taken from gold-standard file, is sense 2.

We can generate three training examples for each sense of the noun *money*. These training examples are presented in Table 1.

Attribute	Training example for Sense 1	Training example for Sense 2	Training example for Sense 3
POS (Part of Speech)	NOUN	NOUN	NOUN
HYPER (Same hierarchy)	UNKNOWN	UNKNOWN	UNKNOWN
PATTERNS (Patterns)	CORRECT	INCORRECT	INCORRECT
DOMAINS (Domains)	INCORRECT	CORRECT	INCORRECT
LEXPAR (Lexical par.)	CORRECT	CORRECT	CORRECT
S_PREV (SemCor prev. word)	CORRECT	INCORRECT	INCORRECT
S_NEXT (SemCor next word)	UNKNOWN	UNKNOWN	UNKNOWN
GDIST (Gloss distance)	INCORRECT	INCORRECT	CORRECT
CREF (Cross reference)	UNKNOWN	UNKNOWN	UNKNOWN
RCREF (Rev.Cross Reference)	UNKNOWN	UNKNOWN	UNKNOWN
FSENSER (First Sense Restr.)	UNKNOWN	UNKNOWN	UNKNOWN
Sense Classification	INCORRECT	CORRECT	INCORRECT

Table 1: Training examples generated for each sense of the noun *money*

We use these training examples in two ways:

STEP 1. The training examples are extracted from a disambiguated gold standard file. Therefore for each training example we have its correct classification. We extract rules that can be applied to new training examples that are not classified.

STEP 2. The training examples are extracted from raw text, therefore we do not have their correct classification. We use the rules learned at STEP 1 to assign a label to each training example: CORRECT or INCORRECT.

Results

First, we measured the accuracy and recall of each method on a set of 3196 gold standard glosses. The *recall* is defined as the ratio between the number of correctly disambiguated words and the total number of target words. The *precision* is defined as the ratio between the number of correctly disambiguated words and the number of attempted words. When we computed these ratios, we did not include the number of monosemous words since these words do not require any effort for disambiguation. The results are presented in Table 2.

Method	Recall	Precision
Same Hierarchy	23.77%	94.48%
Patterns	8.31%	82.66%
Domains	21.48%	81.57%
Lexical Parallelism	12.68%	73.07%
SemCor previous word	3.23%	63.37%
SemCor next word	5.28%	62.39%
Gloss distance	12.10%	63.44%
Cross Reference	4.02%	62.35%
Rev. Cross Reference	2.85%	57.07%
First Sense Restricted	28.68%	58.12%

Table 2: Precision and Recall for WSD methods

Second, we combined these method using four strategies: choosing the sense of the best method, summing the contribution of each method for each sense of the target word and machine learning strategies based on rules extracted using C4.5 and decision lists (Yarowsky 2000).

The first strategy used is obvious: for a word that is tagged with senses by several methods, we choose the sense given by the most accurate method.

The second strategy sums all the contributions of the disambiguation methods for each sense of the word, and selects the sense with the greatest value. We define the contribution of each method to be its accuracy. This is a simple version of the strategy used by Rigau, Atserias, & Agirre (1997) for their system and represents the second approach for combining the classifiers that formed the Stanford WSD system in SENSEVAL-2 competition.

The machine learning strategies consist in C4.5 rules and decision lists. We considered decision lists of size 1, 2, 3 and 4. For the size 1 decision list we considered as a feature an attribute from generated training examples and for each feature f_i we computed a smoothed log-likelihood ration $\frac{P(f_i|s)}{P(f_i|\neg s)}$ (Yarowsky 2000), where s means that the current sense is CORRECT and $\neg s$ means that the current sense is incorrect. We sorted all the features in decreasing order of this ratio and build a decision list.

For decision lists of size 2 we considered as a feature two attributes from generated training examples. For all possible features we computed the same smoothed log-likelihood ratio and built a decision list with the set of features sorted in decreasing order of the ratio value. In the same way we built decision lists for sizes 3 and 4 using as features a combination of 3, respectively 4 attributes.

After generating the rules, for each training example corresponding to a sense of the target word, we applied the rules to tag that sense as correct or incorrect. We assigned a weight to each sense of the target word as being the right sense. This weight is the rule accuracy if the first matched rule labels the sense as correct or negated accuracy if the rule labels the sense as incorrect. Then the final sense selected is the sense with the greatest weight.

For testing the first two strategies we used the whole gold-standard file of 3196 noun glosses. For testing machine learning strategies we performed a 10-fold cross validation. The results are presented in Table 3. As we can see from the table, the approach of using C4.5 rules and decision lists achieve 2-3% gain in accuracy over selecting the sense given by the best method.

Voting scheme	Recall	Precision
Best method	65.94%	76.04%
Summing	65.18%	75.17%
C4.5 rules	62.81%	78.27%
Decision List 1	68.27%	79.04%
Decision List 2	68.07%	78.90%
Decision List 3	68.27%	79.04%
Decision List 4	68.37%	79.35%

Table 3: Precision for different voting schemes

Conclusion and further work

A word in a given context can be disambiguated by several methods. Each method can be right or wrong. The performance of a WSD system depends on the approach of combining the results from several methods. We presented in this paper a combination approach based on generating training examples for each sense of a word, each training example containing the output of each disambiguating method and its part of speech. Then we extracted rules from these training examples using C4.5 and decision lists and used these rules for combining the method for selecting the correct sense. We tested this approach for semantic disambiguation of WordNet glosses. The system gained 3% in precision when compared against the approach that select the sense given by the best method. This approach has the advantage that it can incorporate others sources of information including methods that only say if a given sense of a word in a context is wrong. We can further investigate other learning mechanism from the generated training examples or insert new features in the training example for increasing the accuracy.

References

Harabagiu, S.; Miller, G.; and Moldovan, D. 1999. WordNet 2 - a morphologically and semantically enhanced resource. In *Proceedings of SIGLEX-99*, 1–8.

Klein, D.; Toutanova, K.; Ilhan, H. T.; Kamvar, S. D.; and Manning, C. D. 2002. Combining heterogeneous classifiers for word sense disambiguation. In *Proceedings of the SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, 74–80. Philadelphia: Association for Computational Linguistics.

Magnini, B., and Cavaglia, G. 2000. Integrating subject field codes into WordNet. In *Proceedings of the LREC-2000*.

Mihalcea, R., and Moldovan, D. 2000. An iterative approach to word sense disambiguation. In *Proceedings of FLAIRS-2000*, 219–223.

Mihalcea, R., and Moldovan, D. 2001. eXtended WordNet: progress report. In *Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*, 95–100.

Mihalcea, R. 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*.

Miller, G.and Chodorow, M.; Landes, S.; Leacock, C.; and Thomas, R. 1994. Using a semantic concordance for sense identication. In *Proceedings of the ARPA Human Language Technology Workshop*, 240–243.

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

Novischi, A. 2002. Accurate semantic annotations via pattern matching. In *Proceedings of Florida Artificial Intelligence Research Society*, 375–379.

Quinlan, J. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Rigau, G.; Atserias, J.; and Agirre, E. 1997. Combining unsupervised lexical knowledge methods for word sense disambiguation. In Cohen, P. R., and Wahlster, W., eds., *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, 48–55. Somerset, New Jersey: Association for Computational Linguistics.

Yarowsky, D. 1992. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings, COLING-92*, 454–460.

Yarowsky, D. 2000. Hierarchical decision lists for word sense disambiguation. *Computer and the Humanities* 34(1/2).