Significance of Syntactic Features for Word Sense Disambiguation

Ala Sasi Kanth and Kavi Narayana Murthy

Department of Computer and Information Sciences
University of Hyderabad, India
sasi_kanth_a@yahoo.co.in, knmuh@yahoo.com

Abstract. In this paper[1] we explore the use of syntax in improving the performance of Word Sense Disambiguation(WSD) systems. We argue that not all words in a sentence are useful for disambiguating the senses of a target word and eliminating noise is important. Syntax can be used to identify related words and eliminating other words as noise actually improves performance significantly. CMU's Link Parser has been used for syntactic analysis. Supervised learning techniques have been applied to perform word sense disambiguation on selected target words. The Naive Bayes classifier has been used in all the experiments. All the major grammatical categories of words have been covered. Experiments conducted and results obtained have been described. Ten fold cross validation has been performed in all cases. The results we have obtained are better than the published results for the same data.

1 Introduction

A word can have more than one sense. The sense in which the word is used can be determined, most of the times, by the context in which the word occurs. The word *bank* has several senses out of which *bank* as a financial institution and *bank* as a sloping land bordering a river can be easily distinguished from the context. Distinguishing between the senses of *bank* as a financial institution and *bank* as a building housing such an institution is more difficult. The process of identifying the correct sense of words in context is called *Word Sense Disambiguation* (WSD). Homonymy and Polysemy must both be considered. Word sense disambiguation contributes significantly to many natural language processing tasks such as machine translation and information retrieval.

The focus of research in WSD is on distinguishing between senses of words within a given syntactic category, since senses across syntactic categories are better disambiguated through POS tagging techniques. Many researchers have focused on disambiguation of selected target words although there is some recent interest in unrestricted WSD [1, 2].

[1] The research reported here was supported in part by the University Grants Commission under the UPE scheme.

J. L. Vicedo et al. (Eds.): EsTAL 2004, LNAI 3230, pp. 340–348, 2004.

WSD systems often rely upon sense definitions in dictionaries, features of senses (for example, box-codes and subject categories present in Longman's Dictionary of Contemporary English (LDOCE)), entries in bilingual dictionaries, WordNet etc. Dictionaries and other sources do not always agree on the number and nature of senses for given words. For some tasks the fine granularity of senses as given in some dictionaries is not required or may even be counter productive and so methods to merge closely related senses have been explored by some researchers[3].

Both knowledge based and machine learning approaches have been applied for WSD. Lesk [4] used glossaries of senses present in dictionaries. The sense definition which has the maximum overlap with the definitions of the words in the context was taken as the correct sense. Lesk's algorithm uses the knowledge present in dictionaries and does not use any sense tagged corpus for training. On the other hand, machine learning methods require a training corpus. Yarowsky [5] devised an algorithm which takes some initial seed collocations for each sense and uses unsupervised techniques to produce decision lists for disambiguation. In supervised disambiguation, machine learning techniques are used to build a model from labeled training data. Some of the machine learning techniques used for WSD are - decision lists [6, 7], Naive Bayes classifier [8] and decision trees. In this work we have used the Naive Bayes Classifier.

It has been argued that the choice of the right features is more important than the choice of techniques for classification [9, 10]. A variety of features have been used, including bigrams, surface form of the target word, collocations, POS tags of target and neighboring words and syntactic features such as heads of phrases and categories of phrases in which the target word appears. Some researchers believe that lexical features are sufficient while others [11, 12] have argued for combining lexical features with syntactic features. In this paper we show that syntax can significantly improve the performance of WSD systems. We argue that elimination of noise is important - not all words in a given sentence are useful for disambiguating the sense of a target word. We have used CMU's Link parser to identify words that are syntactically related to the target word. Words which are not syntactically related to the target word are considered to be noise and eliminated. The results we get are comparable to, or better than, the best results obtained so far on the same data.

2 Role of Syntax in WSD

Not all words in the context are helpful for determining the sense of a target word. Syntax can help in identifying relevant parts of the context, thereby eliminating noise. Using syntactic features for WSD is not entirely new. Ng [13] used syntactic information such as verb-object and subject-verb relationships along with the basic lexical features. Yarowsky [14] also used similar syntactic information including verb-object, subject-verb and noun-modifier. Stetina [15] and some of the work presented in the Senseval-2 workshop [16] have also explored the possibility of combining lexical and syntactic features. Recently, Mohammad and Pedersen [11] have analyzed the role of various kinds of lexical and syntactic features in isolation as well as in various combinations. They also employ an ensemble technique to combine the results of classifiers using different sets of features. They propose a method to estimate the best possible performance through such an ensemble technique. They use a simple ensemble method and show that their results are comparable to the best published results and close to the optimal. However, the exact contribution of syntax is not very clear from these studies. David Martinez et al [12] have compared the performance obtained by using only basic (lexical and topical) features with the performance obtained by using basic features combined with syntactic features. They show a performance gain of 1% to 2% for the AdaBoost algorithm while there was no improvement for the Decision List method. In this paper we explore the role of syntactic features in WSD and show that syntax can in fact make a significant contribution to WSD. We have obtained 4% to 12% improvement in performance for various target words. The results we get are comparable to, or better than, the best published results on the same data.

We have used the Link parser developed by Carnegie-Mellon University. The link parser gives labeled links which connect pairs of words. We have found this representation more convenient than parse trees or other representations given by other parsers. Our studies have shown that eliminating noise and using only selected context words is the key to good performance. Syntax has been used only for identifying related words. In the next section we describe the experiments we have conducted and the results obtained.

3 Experimental Setup

Here we have applied supervised learning techniques to perform word sense disambiguation on selected target words. All the major grammatical categories of words have been covered. The Naive Bayes classifier has been used as the base. Ten fold cross validation has been performed in all cases. We give below the details of the corpora and syntactic parser used and the details of the experiments conducted.

3.1 Corpora

For our experiments we have used publicly available corpora converted into Senseval-2 data format by Ted Pedersen [2] We have chosen *interest, serve*, and *hard* as the target words, covering the major syntactic categories - noun, verb and adjective respectively.

In *interest* corpus each instance of the word *interest* is tagged with one of six possible LDOCE senses. There is a total of 2368 occurrences in the sense tagged corpus, where each occurrence is a single sentence that contains the word *interest*. The instances in the corpus are selected from Penn Treebank Wall Street Journal Corpus(ACL/DCI version). Sense tagging was done by Rebecca Bruce

[2] http://www.d.umn.edu/˜tpederse/data.html

and Janyce Wiebe [17]. The sense tags used, frequency, and glosses of the senses are given in Table 1.

Table 1. Senses of the word *interest*, their distribution in the corpus and the gloss of senses

Sense Label	Frequency	Sense Definition
interest_1	361(15%)	readiness to give attention
interest_2	11(01%)	quality of causing attention to be given to
interest_3	66(03%)	activity, etc. that one gives attention to
interest_4	178(08%)	advantage, advancement or favor
interest_5	500(21%)	a share in a company or business
interest_6	1252(53%)	money paid for the use of money

The *hard* corpus contains the word *hard* with part of speech as adjective and is manually tagged with three senses in 4333 contexts. The *hard* data was created by Leacock, Chodorow and Miller [18]. The instances were picked from the San Jose Mercury News Corpus and manually annotated with one of three senses form WordNet. The sense tags used, frequency in the corpus, glosses of the senses and examples are given in Table 2.

Table 2. Senses of the word *hard*, their distribution in the corpus, glosses of senses and examples

Sense Label	Frequency	Sense Definition	Example
HARD1	3455(80%)	not easy - difficult	it's hard to be disciplined
HARD2	502(11%)	not soft - metaphoric	these are hard times
HARD3	376(9%)	not soft - physical	the hard crust

The *serve* corpus contains the word *serve* with part of speech as verb and is manually tagged in 4378 contexts. The *serve* data was created by Leacock, Chodorow and Miller [18]. The instances were picked from the Wall Street Journal Corpus(1987-89) and the American Printing House for the Blind (APHB) corpus. The sentences have been manually tagged with the four senses from WordNet. The sense tags used, frequency in the corpus, glosses of the senses and examples are given in Table 3.

3.2 Parser

For obtaining the syntactic information we have used the link parser from Carnegie-Mellon University[3]. Link parser is a syntactic parser based on link grammar, a theory of English syntax [19, 20]. It is a robust and broad coverage parser. If in case it is unable to parse the sentence fully, it tries to give a partial

[3] http://www.link.cs.cmu.edu/link/

Table 3. Senses of the word *serve*, their distribution in the corpus, and the gloss of senses

SenseLabel	Frequency	Sense Definition	Example
SERVE2	853(20%)	function as something	serves as yard stick to
SERVE6	439(10%)	provide a service	department will serve select few
SERVE10	1814(41%)	supply with food/means	serve dinner
SERVE12	1272(29%)	hold an office	served as head of department

structure to the sentence. Given a sentence, the parser assigns to it a syntactic structure, which consists of a set of labeled links connecting pairs of words. For example the parsed structure of the sentence

```
"The flight landed on rocky terrain"
```

is given by the link parser as

```
                       +------Jp------+
 +--D*u-+---Ss---+-MVp-+     +----A---+
 |      |        |     |     |        |
the flight.n landed.v on rocky.a terrain.n
```

```
A[4]- connects attributive adjectives to following nouns
S - connects subject nouns to finite verbs
D - connects determiners to nouns
J - connects prepositions to their objects
MV- connects verbs and adjectives to modifying phrases that follow,
  like adverbs, prepositional phrases, subordinating conjunctions,
  comparatives and participle phrases with commas.
```

A word to be disambiguated may be connected directly to another word with a link or it can be indirectly connected with a series of links. Consider the target word *interest* in the context:

```
      +---------A--------+
      |         +---AN---+
      |         |        |
....heavy.a interest.n rates.n

AN - connects noun-modifiers to
     following nouns
```

Here the word *rates* is directly linked with the word *interest*, and the word *heavy* is indirectly linked to *interest*. The words which are directly or indirectly

[4] The lower case letters in the label specify additional properties of the links. These are not used in our experiments.

connected to the target word can be taken as the values of the attribute labeled with the name or names of the links. An exclamation mark indicates a leftward link to the target word. The attribute and the value are represented as (attribute,value) pairs. See examples below.

```
    +---AN---+         +---AN---+
    |        |         |        |
interest.n rates.n   key.n  interests.n
    (!AN,rates)           (AN,key)
```

```
  +---------A--------+
  |         +---AN---+
  |         |        |
heavy.a interest.n rates.n
 (!AN,rates)(A-!AN,heavy)
```

Alternatively, we can simply use the syntactically related words as features, without regard to the specific syntactic relationships. The Link parser employs a large number of different types of links and including the link labels greatly increases the space of possible feature values, thereby introducing sparseness in available data. In our experiments described here, we have used only words as features, not the link labels.

4 Experiments and Results

We give below the details of the experiments we have conducted. The results are summarized in the table 4 below.

4.1 Experiment A: Baseline

The baseline performance can be defined as the performance obtained when the most frequent sense in the corpus is assigned for the target word in all contexts. This can be viewed as a bottom-line for comparison of various techniques. The base line performance depends on the flatness of the distribution of the various senses for a given target word. If the senses are all more or less equally likely, the baseline would be low and if one of the senses is much more frequent than others, the baseline would be high. It can be seen that the baseline for *hard* is quite high.

4.2 Experiment B: NaiveBayes (All Words in the Context)

Here all the words in the context of the target word are taken as features. By context we mean the whole sentence in which the target word appears. The words in the context are considered as a bag of words without regard to word order or syntactic structure.

It may be noted that the performance is in general much better than the baseline. (There is, however, a small decrease in the case of the word *hard* - the distribution of senses of the word *hard* is quite peaked and the baseline itself is quite high.)

Not all words in the sentence are likely to be useful for disambiguating the target word. Some words may even have negative effect on the performance. Eliminating or reducing noise should help to achieve better results. The question that remains is the basis for including or excluding words in the context. Syntax captures the internal structure of sentences and explicates a variety of relationships amongst words. We argue, therefore, that syntax should be useful for deciding which words in the sentence are related to the target word and hence likely to influence its sense. The following subsections show various experiments we have carried out to verify this claim. We have found CMU's Link Parser to be an appropriate choice since it directly indicates relationships between pairs of words. Our studies show that syntax indeed has a very significant contribution in improving the performance of WSD.

4.3 Experiment C: NaiveBayes (Syntactically Relevant Words)

In this experiment, all the words which are linked directly or indirectly (up to two levels) to the target word, that is, a bag of selected words from the sentence, are taken as features. It may be seen from the table of results that performance has significantly improved. This vindicates our claim that not all words in context are useful and elimination of noise is important.

4.4 Experiment D: NaiveBayes (Words in a Window)

Our studies have shown that neighboring words often have a considerable influence on the sense of a target word. For example, adjectives and the nouns they modify occur close together and tend to influence one another. The object of a verb may appear close to the verb and have a say in the sense of the verb. Our results for a window of ±2 words around the target word validate this point. The results also confirm our claim that not all words in the sentence are relevant and eliminating noise helps.

We have conducted experiments with various sizes of windows around the target word. The best results are obtained for a window size of 2 to 3. As the window size gets larger, more and more noise words will start getting in and the performance drops.

4.5 Experiment E: NaiveBayes (Syntactically Related and Neighborhood Words)

Here we try to combine syntactically linked words and words in the neighborhood of the target word for feature selection. Note that neighboring words may not always be linked syntactically. In this study words syntactically related to the target word either directly or through one level of intermediate link have been included. Also, words in a neighborhood of ±2 are included. All other words in the sentence are treated as noise and ignored. It may be seen that this combination generally outperforms the other schemes. In fact the results obtained here are better than the best published results [11]. It may be noted that we have actually used only lexical features, not any syntactic features directly. Syntax has been used only to identify related words and remove other words as noise.

We have also conducted experiments where the features included not just the related words but also the specific syntactic relations as expressed by the links in the Link parser. This greatly enlarges the feature space as there are several hundred different types of links. The sparseness of training data under this expanded feature space will limit the performance obtainable. Our experiments have shown that not much improvement in performance is possible with the available data.

4.6 Results

Table 4 shows the results of our experiments:

Table 4. Table showing the accuracy (in %) for the three words

words	Experiment				
	A: Baseline	B: All words in sentence	C: Only syntactically related words	D: Neighboring words	E: Selected words
interest(n)	52.87	85.66	86.14	87.94	89.39
hard(ad)	79.73	78.22	90.59	91.53	90.91
serve(v)	41.43	76.88	81.93	81.23	85.25

5 Conclusions

In this paper we have explored the contribution of syntax for word sense disambiguation. Not all words in a sentence are helpful for identifying the intended sense of a given target word. Syntactic information can be used to identify useful parts of the context and thereby reduce noise. We have not directly used any syntactic features. Syntax helps in the selection of the right lexical features and our experiments show that elimination of noise can significantly improve the performance of WSD. Improvement in performance ranges from about 4 % to about 12 %. Overall performance achieved ranges from about 85 % to about 90 % and is comparable to, or better than, the best results published on similar data.

References

1. Mihalcea, R., Moldovan, D.: A method for word sense disambiguation of unrestricted text. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99), Maryland, NY (1999)
2. Wilks, Y., Stevenson, M.: Word sense disambiguation using optimized combinations of knowledge sources. In: Proceedings of ACL 36/Coling 17. (1998) 1398–1402

3. Dolan, W.B.: Word sense ambiguation: Clustering related senses. Technical Report MSR-TR-94-18, Microsoft Corporation, Redmond, WA (1994)
4. Lesk, M.: Automated sense disambiguation using machine-readable dictionaries: How to tell a pine cone from a ice cream cone. In: proceedings of the SIGDOC Conference, Toronto, Canada (1986) 24–26
5. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: Meeting of the Association for Computational Linguistics. (1995) 189–196
6. Rivest, R.: Learning decision lists. Machine Learning **2** (1987) 229–246
7. Yarowsky, D.: Decision lists for lexical ambiguity resolution: application to accent restoration in spanish and french. In: Proceedings of ACL '94. (1994) 88–95
8. Gale, W.A., Church, K.W., Yarowsky, D.: Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In: Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, University of Delaware, Newark, Delaware (1992) 249–256
9. Pedersen, T.: A decision tree of bigrams is an accurate predictor of word sense. In: Proceedings of the Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics. (2001) 79–86
10. Ng, H.T., Lee, K.L.: An empirical evaluation of knowledge sources and learning algirithms for word sense disambiguation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2002) 41–48
11. Mohammad, S., Pedersen, T.: Combining lexical and syntactic features for supervised word sense disambiguation. In: Proceedings of CoNLL-2004, Boston, MA, USA (2004) 25–32
12. Martínez, D., Agirre, E., Màrquez, L.: Syntactic features for high precision word sense disambiguation. In: Coling. (2002)
13. Ng, H.T., Lee, H.B.: Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In: Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, San Francisco, Morgan Kaufmann Publishers (1996) 40–47
14. Yarowsky, D.: Hierarchical decision lists for word sense disambiguation. Computers and the Humanities **34** (2000)
15. Stetina, J., Kurohashi, S., Nagao, M.: General word sense disambiguation method based on a full sentential context. In: Usage of WordNet in Natural Language Processing, Proceedings of COLING-ACL Workshop, Montreal, Canada. (1998)
16. J, P., Yarowsky, D. In: Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (Senseval 2). In conjunction with ACL2001EACL2001, Toulouse, France. (2001)
17. Bruce, R., Wiebe, J.: Word-sense disambiguation using decomposable models. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics. (1994) 139–146
18. Leacock, Chodorow, Miller: Using corpus statistics and wordnet relations for sense identification. In: Computational Linguistics. Volume 24(1). (1998)
19. Sleator, D., Temperley, D.: Parsing english with a link grammar. Technical Report CMU-CS-91-196, Carnegie Mellon University (1991)
20. Sleator, D.D., Temperley, D.: Parsing English with a link grammar. In: Third International Workshop on Parsing Technologies. (1993)