Article Selection Using Probabilistic Sense Disambiguation

Lee Hian-Beng

DSO National Laboratories
20 Science Park Drive, Singapore 118230

Abstract

A probabilistic method is used for word sense disambiguation where the features taken are the surrounding six words. As their surface forms are used, no syntactic or semantic analysis is required. Despite its simplicity, this method is able to disambiguate the noun *interest* accurately. Using the common data set of (Bruce & Wiebe 94), we have obtained an average accuracy of 86.6% compared with their reported figure of 78%. This portable technique can be applied to the task of English article selection. This problem arises from machine translation of any source language without article to English. Using texts from the Wall Street Journal, we achieved an overall accuracy of 83.1% for the 1,500 most commonly used head nouns.

1 Introduction

Word sense disambiguation is an important problem in natural language processing, which has been addressed in a variety of ways. A popular strategy is the knowledge-based approach, in which linguistic knowledge is used to achieve disambiguation. For example (Agirre &: Rigau 95) have made use of the WordNet noun hierarchy to select the word sense which shares the same WordNet noun subtree as most of the surrounding nouns. (Véronis & Ide 95) took advantage of machine-readable dictionaries instead, identifying similarities between words through definitions in the dictionary. The thesaurus has also been used by Yarowsky 92) and (Okumura & Honda 94), whose semantic categories serve as classes for disambiguation. (Wilks & Stevenson 98) used knowledge such as dictionary definitions, pragmatic codes and selectional restrictions for word sense disambiguation. However, these approaches all require a WordNet-like lexical knowledge source or a machine-readable dictionary, which often take years to construct. Relying on such resources also makes the extension of these methods to other languages difficult.

Recently, much attention has been focused on the statistical approach to solving the problem. In particular, (Bruce & Wiebe 94) have used a probabilistic classifier based on keywords, as well as syntactic information such as morphology and parts of speech. (Yarowsky 92) has modeled the sense-disambiguation problem using the naive Bayesian classifier, where the features used are words in the surrounding 100-word window. Instead of this set of features, (Leacock et al. 93) have chosen to use two-sentence contexts: the sentence containing the word to be disambiguated and the preceding sentence. Although humans appear to require only a few words for sense resolution (Choueka & Lusignan 85), large window sizes have been found to be necessary in these studies presumably because so much information, such as word-order and syntax, has been thrown away. (Mooney 96) has also studied the Bayesian classifier and compared it with other methods, although in the preprocessing he reduces words to stems and removes stopwords.

In this paper, we present a probabilistic disambiguation algorithm which does not require any syntactic or semantic information (Teo et al. 97). The features used are the surrounding six words, in their surface form and taking into account their relative positions. This is to be contrasted with the studies of (Yarowsky 92) and (Leacock et al. 93) mentioned above, which use a very large unordered window of words. As we will demonstrate, this minimal-knowledge approach to word sense disambiguation is able to achieve a high accuracy compared with other methods. The simplicity of our approach also makes it generic: it can be adapted for use with other languages without having to modify the engine. It can also be applied to a wide range of classification problems. In machine translation for example, when a word *W* of the source language has a number of possible translations in the target language, one can use this technique to determine the correct translation from the neighboring context of *W*.

We have tested the word sense disambiguation algorithm on the noun *interest,* using a corpus with 2,369 word occurrences made publicly available by (Bruce & Wiebe 94). We believe it is important to use

standard data sets to measure disambiguation performance.

However, the evaluation of this technique on only one word can hardly be indicative of its general usefulness. We have thus further tested it on a large corpus consisting of the most frequently occurring 121 nouns and 70 verbs in the Brown corpus and Wall Street Journal corpus. This data set was first used by (Ng & Lee 96).

We applied this disambiguation method to article selection for English output generated by machine translation system. Many languages, such as Chinese, and Thai do not have articles. However, English language uses articles. Yet, this does not mean that the articles in English language are redundant. Articles such as "a/an" and "the" carry semantic information. The use and the choice of articles are important for native English speakers. Article-free English text is difficult to read.

We tested the algorithm for article selection using texts from the Wall Street Journal (WSJ). For the task of selecting between "a/an" and "the", we achieved an accuracy of 83.1% for the 1,500 most frequently used head nouns.

2 Sense Disambiguation Algorithm

The probabilistic word sense disambiguation (PWSD) technique we used requires a set of feature tables containing the feature entries and their frequency counts, as extracted from a training corpus. The features are *position-dependent* surface words[1], which are within a certain vicinity of the word to be disambiguated in the sentences. We denote the feature corresponding to the i-th word to the left by f^{Li}, and that corresponding to the i-th word to the right by f^{Ri}.

As only surface words are used, we do not require part-of-speech tagging, morphological analysis, parsing or any kind of syntactic analysis. In other words, minimal preprocessing is required, and is restricted to sentence boundary disambiguation and tokenization.

Let W be a polysemous word with N classes. To construct the feature tables, we require a training corpus containing instances of W being tagged as sense $J = 1,\ldots,N$ in the contexts where it occurs. The features are extracted from the surrounding n words to the left and n words to the right. For each feature (word position), all the possible words are stored in the feature table, together with their frequency counts and the conditional probabilities of them being used with sense J. Having prepared these tables, we disambiguate the sense of W in a test sentence as follows:

[1] By 'word', we mean the tokens in a sentence. No distinction is made between upper and lower cases. The only word class we have is **NUM**, which replaces the surface words if they are numbers.

Calculate the score of each sense J:

$$S(J) = \sum_i \beta^i \, \mathrm{P}(J|f^i) \,. \qquad (1)$$

Choose the sense of W:

$$\text{Sense of } W = \arg\max_J S(J) \,. \qquad (2)$$

In (1), $i = \mathrm{L}1,\ldots,\mathrm{L}n,\mathrm{R}1,\ldots,\mathrm{R}n$ ranges over the $2n$ features extracted from the test sentence. The conditional probabilities $\mathrm{P}(J|f^i)$ of sense J given feature f^i can be read off from the appropriate feature table, and is set to zero if not found in the table. The importance of feature i relative to the other features is given by the scalar weight β^i. It is reasonable to assume that features closer to W are more important than those farther away, and so the former should have larger β^i's. The score of sense J is the weighted sum of these conditional probabilities. The sense of W is then chosen to be $\hat{J}$ such that $S(\hat{J})$ is the highest score.

If none of the features match, all the scores remain zero. We then, by default, choose the most frequently-used sense.

In the above discussion, we did not fix a value for the size n of the window of features. Neither did we specify the weights β^i. The optimal choice for these values would no doubt vary from word to word. Here, we have tried out several different possibilities and adopted values which give the best accuracy for the large corpus of 121 nouns and 70 verbs mentioned in the introduction.

The first case we tried was to fix all the weights to be unity (so that all the features contribute equally to the decision process) and vary the window size. We have found that the best results are obtained when $n = 2$, supporting the earlier conjecture that a small window size is adequate for disambiguation.

The next case we tested out was to vary the weights with the distance from W. In particular, we set

$$\beta^{\mathrm{L}n} = \beta^{\mathrm{R}n} = 2^{1-n}, \qquad (3)$$

which decreases exponentially with n. It was found that $n = 3$ is optimal, with an overall accuracy about 1% better than the case with $n = 2$ and all weights equal. Thus our final choice is for a size $n = 3$ and weights $\beta^{\mathrm{L}1} = \beta^{\mathrm{R}1} = 2.0$, $\beta^{\mathrm{L}2} = \beta^{\mathrm{R}2} = 1.0$, and $\beta^{\mathrm{L}3} = \beta^{\mathrm{R}3} = 0.5$. Note that we are free to scale the weights by an arbitrary amount without affecting the results. This set of weights is consistent with the observation by (Yarowsky 95), that words farther away are less important.

We have tested the disambiguation algorithm on the noun *interest*, which has the six senses listed in Table 1. Training sets of various sizes were randomly extracted from the 2,369-sentence data set of (Bruce & Wiebe 94), in proportion with the overall class distribution. Statistics were then gathered from the training set, and the disambiguation routine tested on the

Table 1: Sense-tag distribution of the word *interest*

No.	Sense	Sentences	Percentage
1	"readiness to give attention"	361	15%
2	"quality of causing attention to be given"	11	< 1%
3	"activity, subject, etc., which one gives time and attention to"	66	3%
4	"advantage, advancement or favor"	178	8%
5	"a share (in a company, business, etc.)"	500	21%
6	"money paid for the use of money"	1253	53%

remaining sentences. This was averaged over 100 times for a fixed number of training sentences, and the results are plotted in Figure 1.

Notice that there is an initial phase when disambiguation accuracy increases rapidly with the size of the training set. However, this increase starts leveling off when we reach around 1000 training sentences. The asymptotic accuracy in the limit of infinite training sentences appears to be about 90%.

With 1769 training sentences (600 left for testing), our disambiguation algorithm achieves an average accuracy of 86.6%. This is almost 9 percentage points higher than the figure of 78% reported by (Bruce & Wiebe 94) for a similarly sized training set. It is comparable to that recently reported by (Ng & Lee 96), using the approach of nearest neighbors.

To further evaluate our disambiguation algorithm, we have tested it on the large corpus that was used by LEXAS (Ng & Lee 96). It consists of 192,800 word occurrences, of which 113,000 are occurrences of 121 nouns, and 79,800 are occurrences of 70 verbs. There are an average of about 1000 examples for each word to be disambiguated. These sentences were drawn from the 1-million-word Brown corpus and the 2.5-million-word Wall Street Journal corpus. The senses are taken from WordNet 1.5, with an average of 7.8 senses per noun and 12.0 senses per verb.

Two different subsets were separately used for testing. The first set, named BC50, consists of 7,119 occurrences of the 191 words in 50 selected text files of the Brown corpus. The second set, named WSJ6, consists of 14,139 occurrences of the 191 words in six selected text files of the Wall Street Journal corpus. The proportion of the data set aside for testing is about 11%. The disambiguation accuracy (in percentage) on these two test sets are tabulated below:

Test set	Baseline	LEXAS	Ng 97	PWSD
BC50	47.1	54.0	58.7	59.0
WSJ6	63.7	68.6	75.2	74.9

Our results in the last column were comparable to those obtained by (Ng 97). The figures shown were his best results for the case of using 10-fold cross validation to select the best k value (for exemplar-based method). When compared to the default strategy of picking the most frequent sense in the training data, the improvement ranges between 11 and 12 percentage points. Thus, this disambiguation algorithm is able to perform well even on a large set of words. Note that the accuracy attained on the Brown corpus is lower than that achieved on the Wall Street Journal corpus, because the former consists of texts from a wider variety of domains.

It is necessary to understand why it works well after we demonstrated our model for word sense disambiguation. The first reason can be traced to our use of surface words as features, which is obviously more precise than the parts of speech that (Bruce & Wiebe 94) mostly use. The second is the small window size that has been adopted. Our experiments showed that too large a window size does not enhance the disambiguation accuracy. The optimal turns out to be about two or three words to the left and to the right of the word to be disambiguated, as can be seen from the following accuracy figures:

Test set	n = 1	2	3	4	5
BC50	57.6	58.3	57.8	56.8	56.1
WSJ6	73.6	74.3	73.5	72.9	72.5

Finally, adjusting the weights of the different features showed that features closer to the word are generally more important in the decision process than those farther away. Using the formula for /?* in (3), we have:

Test set	n = 1	2	3	4	5
BC50	57.6	58.9	59.0	58.9	58.9
WSJ6	73.6	74.8	74.9	74.8	74.8

The best case, as used for reporting the results in the previous section, corresponds to n = 3, although the other cases are not too far behind. As all the experiments were performed on the large corpus of 191

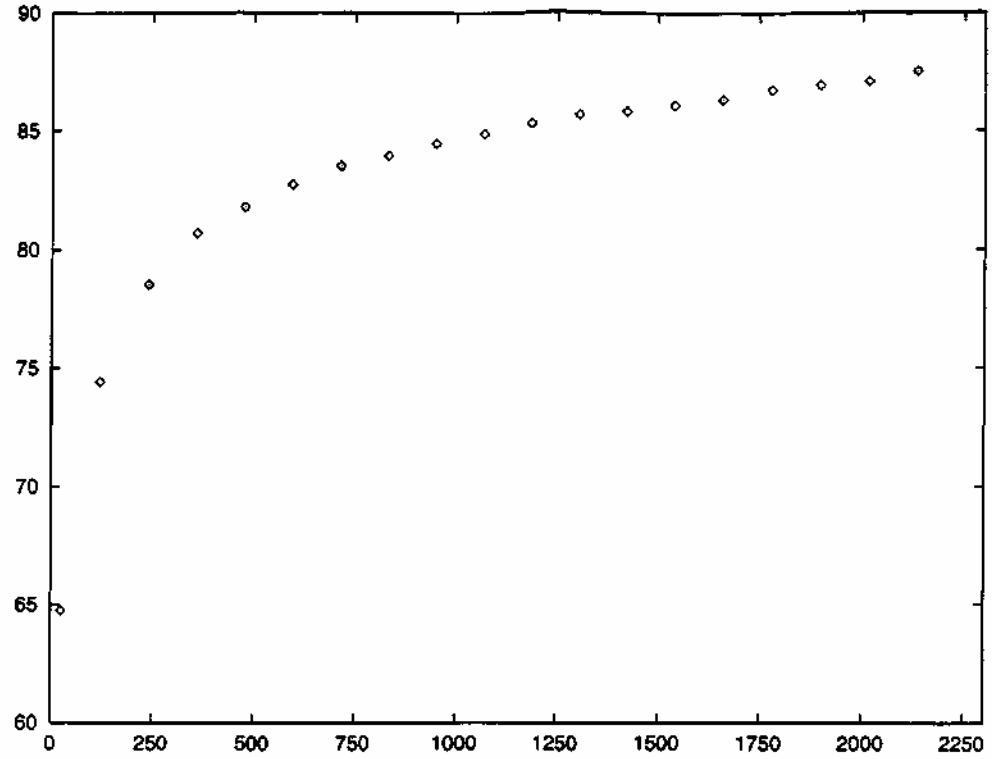

Figure 1: Disambiguation accuracy of the test set versus the number of training sentences for the noun *inter*

most frequently occurring nouns and verbs, we would expect this model to continue to do well in most other situations.

The simplicity of our algorithm also makes it practical. Not only is it fast (about 500 examples processed per second on a Silicon Graphics workstation with an R4400 processor), it can be easily adapted to other languages and even other classification problems, without having to modify the engine.

3 Article Selection

Our PWSD algorithm is highly portable because minimum knowledge (only the surface words) is used. We apply it to the task of English article selection. Article selection can be viewed as a sense disambiguation problem. For each head noun in a translated noun group, we want to decide the best article based on the context. Three selections are possible: no article, "the", or "a/an". Note that articles "a" and "an" can be considered as a single class because they are the same as far as semantic is concerned. We could distinguish them by using a lookup of a list of words starting with vowel sounds. Alternatively, we can also use PWSD to select article from these four possibilities.

The training examples can be collected from a raw English corpus. We used the Wall Street Journal (WSJ) to test our PWSD algorithm for article selection. For each head noun, we have to collect enough training examples for the PWSD algorithm to perform. In applying the algorithm, we use the article as the centre of the window of words. Because of the notion of head noun and the need to recognize the right article for this head noun, we need a noun phrase parser (Ting 95) and a POS tagger for English language. The accuracy of the two programs are 97% and 96% respectively. Collection of the training examples for "a", "an" and "the" can be done automatically. No human tagging is required. We do not consider examples with other determiners such as "this", "that", "any".

The same problem of selecting English article is encountered in JAPANGLOSS system. (Knight and Chander 1994) have reported similar work using a selection method based on decision trees. For the 1,600 most popular head nouns (which have sufficient examples in WSJ), they achieved 81% accuracy for the selection of two classes of articles − "the" and "a/an". For the remaining head nouns, a default strategy of using "the" is adopted. The overall accuracy was 78%.

We performed a similar test for comparison. We set aside three WSJ texts (WSJ3) for final testing for the overall accuracy. We have 8900 test examples. We collected training examples from the rest of WSJ texts

for the 1,500 most commonly used head nouns. For these nouns, we achieved an overall accuracy of 83.1%. These nouns cover about 77.1% of the test examples. The most frequent selection for these test examples has an accuracy of 64.7%. We have an improvement of 18.4% for the nouns with enough examples (which range from a few dozens to a few thousands). The remaining test examples were given a default of "the". The overall accuracy was 81.2%.

Table 2 shows the disambiguation results for the 20 most popular head nouns. The test was performed by collecting all the examples from the WSJ. Training was performed on 90% of the data and testing on the remaining 10%. 25 runs using random selection of training and test examples were performed for each noun and the average accuracy was reported. The average accuracy is about 15% higher than that of the most frequent selection. Note that "the" is not always the most frequently used article, nouns marked with * have "a/an" as most likely choice.

Table 2: The performance of article selection using PWSD

noun	baseline %	PWSD %
year	*50.6	94.2
company	86.1	89.5
share	*85.4	98.5
market	91.4	92.6
price	71.3	88.3
sale	87.3	89.4
month	78.3	94.8
stock	89.0	92.2
rate	68.1	86.5
president	56.3	85.7
time	73.8	89.9
week	67.7	90.9
business	84.1	85.4
analyst	*84.1	89.5
day	62.0	90.1
official	*63.9	81.1
issue	82.7	87.5
people	89.9	94.9
investor	52.3	77.0
group	55.5	85.2
average	74.0	89.1

The method can also be used for the selection of three classes of articles − null article, "the" and "a/an". There is *no* change of the algorithm, but only in the collection of examples. We have 27,800 examples from the WSJ3 for testing. For the most popular 1,500 head nouns, we achieved an overall accuracy of 80.3% over a baseline accuracy of 58.1%. We have an impressive improvement of about 22%. For those outside the 1.500 nouns, a default of null article is used. We have an overall accuracy of 81.1%. The widening of the choice of articles does not degrade the performance of PWSD. This could be partly due to the presence of plural noun forms for the case of null article. This type of article selection is more practical because null article is the most likely choice (and hence cannot be omitted). The results for the 20 most frequently used head nouns are given in table 3.

Table 3: The performance of selecting three classes of articles using PWSD

noun	baseline %	PWSD %
year	50.8	91.2
company	58.8	88.4
share	51.2	91.8
market	*69.3	80.8
price	67.0	83.1
sale	72.1	87.7
month	55.8	91.8
stock	73.6	82.7
rate	61.4	81.6
president	74.9	79.9
time	48.5	85.4
week	60.2	88.2
business	66.9	75.2
analyst	71.0	88.3
day	53.4	89.0
official	76.4	83.2
issue	48.2	79.9
people	87.0	88.3
investor	87.6	89.5
group	*37.4	77.5
average	63.6	85.2

Naturally, the method can be extended for selecting four classes of articles − "a", "an", "the" and null article. For the 20 most frequently used nouns, we achieved an average accuracy of 84.6%. The slight degradation of performance is due to the small number of examples for differentiating "a" and "an". These two classes of examples are the smallest groups for most of the nouns. A better solution is not to split "a" and "an" at this stage but to look up later through a collective list of all the words with vowel sounds from all the examples.

It has been shown that PWSD could be used for article selection without any manual tagging. The performance of the algorithm can be enhanced if more training examples can be collected from a larger corpus. This is especially true for those nouns with only a few dozens of training examples. English texts for all kinds of domains can be extracted readily from various Internet web sites.

This algorithm can be used as a post-processor for MT systems to insert articles with the help of an English noun phrase parser. It is also useful in correcting the choice of articles for texts written by non native speakers. They find accurate selection of articles very difficult. An overall accuracy of 81% for selecting three classes of articles simply by using our PWSD

algorithm is reasonable for these applications.

4 Conclusion

In conclusion, we have developed a probabilistic model of word sense disambiguation. Despite our knowledge-lean approach to the problem, it is able to achieve disambiguation accuracies on the high end of the scale for the word *interest*. The algorithm can be applied to the problem of article selection. The required training examples can be collected automatically from raw texts.

References

Agirre E. and Rigau G. (1995). "A proposal for word sense disambiguation using conceptual distance". In proceedings of the 1st International Conference on Recent Advances in Natural Language Processing.

Bruce R. and Wiebe J. (1994). "Word sense disambiguation using decomposable models". In proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics.

Choueka Y. and Lusignan S. (1985). "Disambiguation by short contexts". In Computers and the Humanities, 19:147-157.

Knight K. and Chander I. (1994). "Automated Postediting of Document". In proceedings of the National Conference on Artificial Intelligence (AAAI).

LeacockC., Towell G. and Voorhees E. (1993). "Corpus-based statistical sense resolution". In proceedings of the ARPA Workshop on Human Language Technology.

Mooney R.J. (1996). "Comparative experiments on disambiguating word senses: an illustration of the role of bias in machine learning". In proceedings of the Conference on Empirical Methods in Natural Language Processing.

Ng H.T. (1997). "Exemplar-based Word Sense Disambiguation: Some Recent Improvements". In proceedings of the Conference on Empirical Methods in Natural Language Processing.

Ng H.T. and Lee H.B. (1996). "Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach". In proceedings of the 34th Annual Meeting of the Association for Computational Linguistics.

Okumura M. and Honda T. (1994). "Word sense disambiguation and text segmentation based on lexical cohesion". In proceedings of the 15th International Conference on Computational Linguistics.

Teo E., Lee H.B., Ting C. and Peh C.S. (1997). "Probilistic word sense disambiguation: a portable approach using minimum knowledge". In proceedings of the Second International Conference on Recent Advances in Natural Language Processing.

Ting C. (1995). "DESPAR, A dependency structure parser without using any grammar formalism". In Industrial Parsing of Software Manuals.

Véronis J. and Ide N. (1995). "Large neural networks for the resolution of lexical ambiguity". In Computational Lexical Semantics, Cambridge University Press.

Yarowsky D. (1992). "Word sense disambiguation using statistical models of Roget's categories trained on large corpora". In proceedings of the 15th International Conference on Computational Linguistics.

Yarowsky D. (1992). "Unsupervised word sense disambiguation rivaling supervised methods". In proceedings, 33rd Annual Meeting of the Assn. for Computational Linguistics.

Wilks Y. and Stevenson M. (1998). "Word sense disambiguation using optimized combinations of knowledge sources". In proceedings, 36rd Annual Meeting of the Assn. for Computational Linguistics.