Using WordNet to Disambiguate Word Senses for Text Classification

Ying Liu[1], Peter Scheuermann[2], Xingsen Li[1], and Xingquan Zhu[1]

[1] Data Technology and Knowledge Economy Research Center, Chinese Academy of Sciences
Graduate University of Chinese Academy of Sciences
100080, Beijing, China
yingliu@gucas.ac.cn, lixingsen@126.com, xqzhu@cse.fau.edu
[2] Department of Electrical and Computer Engineering
Northwestern University, Evanston, Illinois, USA, 60208
peters@ece.northwestern.edu

Abstract. In this paper, we propose an automatic text classification method based on word sense disambiguation. We use "hood" algorithm to remove the word ambiguity so that each word is replaced by its sense in the context. The nearest ancestors of the senses of all the non-stopwords in a give document are selected as the classes for the given document. We apply our algorithm to Brown Corpus. The effectiveness is evaluated by comparing the classification results with the classification results using manual disambiguation offered by Princeton University.

Keywords: disambiguation, word sense, text classification, WordNet.

1 Introduction

Text classification aims at automatically assigning a document to a pre-defined topic category. A number of machine learning algorithms have been investigated for text classification, such as K-Nearest Neighbor (KNN) [1], Centroid Classifier [2], Naïve Bayes [3], Decision Trees [4], Support Vector Machines (SVM) [4]. In such classifiers, each document is represented by a *n*-dimensional vector in a feature space, where each feature is a keyword in the given document. Then traditional classification algorithms can be applied to generate a classification model. To classify an unseen document, a feature vector is constructed by using the same set of *n* features and then passed to the model as the input. These methods suffer from the nature of text documents [6]. It is not feasible to organize a document into a fixed set of features because most text documents are semi-structured or completely not structured. An alternative type of approaches, keyword-based association analysis, has been proposed in [13,14,15]. Such classifiers proceed as follows: firstly, keywords and terms are extracted; secondly, concept hierarchies of keywords and terms are obtained by using WordNet, or expert knowledge, or some keyword classification systems. Documents in the training set can be classified into class hierarchies. A term association mining method is proposed to discover sets of associated terms that can be

Y. Shi et al. (Eds.): ICCS 2007, Part III, LNCS 4489, pp. 780–788, 2007.

used to distinguish one class from others. It derives a set of association rules associated with each document class. Such rules can be used to classify new documents.

However, word ambiguity is a severe problem in the keywords-based methods. For example, if 'bat' occurs several times in a document, should the file be classified to "sport" or "mammal"? A number of computer engineers tried to retrieve articles about "board", but a large number of Web pages about "board game" or "message board" were retrieved. Each word may have multiple senses (meanings), and multiple words may have the same sense. It is not trivial for a computer to know which sense the keyword is using in a given context. Extensive research has been done in word sense disambiguation [5,16,17,18]. However, to the best of our knowledge, disambiguation research is focused in retrieval or in query, not for text classification.

In this paper, we propose a text classification method based on sense disambiguation. In order to define an appropriate mid-level category for each sense, hood [5] is implemented on WordNet. Each keyword in a given document is mapped to the concept hierarchy where each sense maintains a counter. The hoods and the associated counters determine the intended sense of a given ambiguous word. Thirdly, the ancestors of the synsets of all the keywords are selected as the classes of a given document. We apply this algorithm to Brown Corpus. The effectiveness of our automatic text classification method is evaluated by comparing the classification results with the classification results using manual disambiguation offered by Princeton University.

The rest of this paper is organized as follows. Section 2 overviews the related work. Section 3 introduces WordNet. In Section 4, we present the sense disambiguation-based text classification algorithm. Section 5 presents our experiment results and discussion. We summarize our work in Section 6.

2 Related Work

Knowledge-based. In this category, disambiguation is carried out by using information from an explicit lexicon or knowledge base. The lexicon may be a machine readable dictionary, thesaurus or hand-crafted. [9,10,5] use WordNet as the knowledge base to disambiguate word senses, and [11] uses Roget's International Thesaurus.

Corpus-based. This category of approaches attempt to disambiguate words by using information gained from training on some corpus, rather than taking it directly from an explicit knowledge source [8]. Training can be carried out either on a disambiguated corpus or a raw corpus. In a disambiguated corpus, the semantics of each polysemous lexical item has been marked, while in a raw corpus, the semantics has not been marked yet.

Hybrid Approaches. A good example is Luk's system [12] which uses the textual definitions of senses from a machine readable dictionary to identify relations between senses. It then uses a corpus to calculate mutual information scores between the related senses in order to discover the most useful information. In this way, the amount of text needed in the training corpus is reduced.

3 WordNet

WordNet is a manually-constructed lexical system developed by George Miller at the Cognitive Science Laboratory at Princeton University [7]. It reflects how human beings organize their lexical memories. The basic building block of WordNet is synset consisting of all the words that express a given concept. Synsets, which senses are manually classified into, denote synonym sets. Within each synset, the senses, although from different keywords, denote the same meaning. For example, "*board*" has several senses, so does "*plank*". Both of them have a common sense "a stout

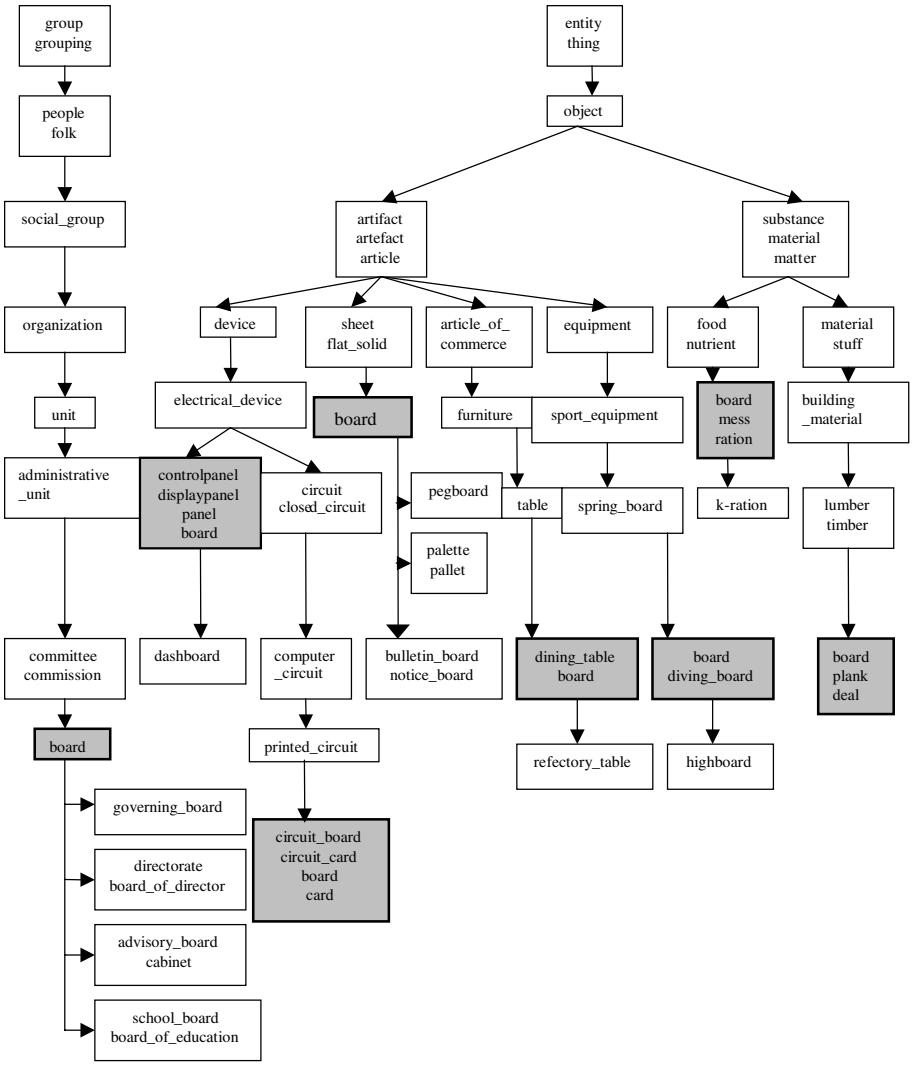

Fig. 1. The IS-A hierarchy for eight different senses of the noun "board"

length of sawn timber, made in a wide variety of sizes and used for many purposes". Thus, "*plank*" and "*board*" are synonymous in terms of this specific sense and form one synset. Because all synonymous senses are grouped into one synset and all different senses of the same word are separated into different synsets, there is no synonymous or polysemous synset separated into different synsets, there is no synonymous or polysemous synset. Hence, WordNet is a concept-based dictionary where every synset represents a lexicalized concept. WordNet consists of four divisions, nouns, verbs, adjectives and adverbs division. Within a division, synsets are organized by the lexical relations defined on them. We only use the noun division of WordNet in this study due to the page limitation. We use two lexical relations in the noun division, "IS-A" and "PART-OF".

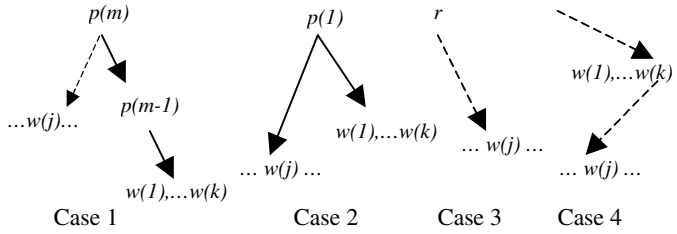

Fig. 2. Root of hoods of synset *s*

Figure 1 shows the hierarchy relating to the eight different senses of noun "*board*". The synsets with the heavy boundary are the actual senses of "*board*", and the remaining synsets are either ancestors or descendents of the senses. The synsets {*group, grouping*} and {*entity, thing*} are examples of heads of the hierarchies. WordNet 1.6 (2000) contains 94473 words and 116314 senses in the noun division. It is used as the framework of our proposed hierarchical text classification method.

4 Word Sense Disambiguation and Text Classification

In this section, we first present our implementation of "hood" proposed in [5] on WordNet. Hood is based on the idea that a set of words co-occurring in a document will determine the appropriate senses for one another word despite each individual word being multiply ambiguous. A common example of this effect is the set of nouns base, bat, glove and hit. Although each of them has several senses, when taken together, the intent is baseball game, clearly. To exploit this idea automatically, a set of categories representing the different senses of words needs to be defined. A counter is maintained in each category, which counts the number of words that have its associated senses. The sense of an ambiguous word is determined by the category with the largest counter. Then, the nearest ancestors of the senses of all the non-stopwords are selected as the classes of a given document.

4.1 Hood Construction

Using each separate hierarchy as a category is well defined but too coarse grained. For example, in Figure 1, 7 of 8 senses of "*board*" are in the {*entity, thing*} hierarchy.

Therefore, *hood* is intended to define an appropriate middle level category. To define the hood of a given synset, *s*, consider the synsets and the hyponymy links in WordNet as vertices and directed edges of a graph. Then, the hood of *s* is defined as the largest connected subgraph that contains *s*, containing only descendents of an ancestor of *s*, and containing no synset that has a descendent that includes another instance of a member of *s* as a member. A hood is represented by the root of the hood. Figure 2 illustrates the definition of hood, assuming synset *s* consists of *k* words w_1, w_2, $w_3 \ldots w_k$, and p_1, p_2, $p_3 \ldots p_n$ are *n* ancestors of *s*, where p_m is a father of p_{m-1}. p_m ($1 \leq m \leq n$) has a descendent synset which also includes w_j ($1 \leq j \leq k$) as a member. So, p_{m-1} is one of the roots of the hoods of *s*, as shown in Case 1. If *m* is 1, *s* itself is the root, shown in Case 2. If no such *m* is found, the root of WordNet hierarchy, *r,* is the root of the hood of *s*, as shown in Case 3. If *s* itself has a descendent synset that includes w_j as a member, there is no hood in WordNet for *s*, as shown in Case 4. Because some synsets have more than one parents, synsets can have more than one hoods. A synset has no hood if the same word is a member of both the synset and one of its descendents. For example, in Figure 1 the hood of synset "*committee sense*" of "*board*" is rooted at synset {*group, grouping*} (and thus the hood for that sense is the entire hierarchy where it occurs) because no other synset containing "*board*" in this hierarchy (case 3); the hood of "*circuit_board*" sense of "*board*" is rooted at {*circuit, closed_circuit*} because synset {*electrical_device*} has a descendent {*control_panel, display_panel, panel, board*} containing "*board*" (case 1), and the hood of "*panel*" sense of "*board*" is rooted at the synset itself because its direct parent {*electrical_device*} has a descendent synset {*circuit_board, circuit_card, board, card*} containing "*board*" (Case 2).

4.2 Word Sense Disambiguation

After the hoods for each synset in WordNet are constructed, they can be used to select the sense of an ambiguous word in a given text-document. The senses of the nouns in a text document in a given document collection are selected by using the following two-step process. A procedure, called *marking* (*w*), is fundamental to both of the steps. *Marking* (*w*) visits synsets and maintains a counter for each synset, which is increased by 1 whenever the synset is visited. Given a word *w*, what *marking* (*w*) does is to find all instances of *w* in WordNet, and then, for each identified synset *s*, follow the parent-child links up to the root of the hierarchy while incrementing the counter of each synset it visits. The first step of the two-step process is collection-oriented, that is, *marking* (*w*) is called for each occurrence of *w* in all the documents in the collection. The number of times *marking (w)* is called for each *w* is maintained by some counters. The first step produces a set of global counts (relative to this particular collection) at each synset. The second step is document-oriented, that is, *marking* (*w*) is called for each occurrence of *w* in an individual text document. Again the number of times *marking (w)* is called is maintained for the given individual document. The second step produces a set of local counts at the each synset. Given the local and global counts, a sense for a given ambiguous word *w* contained within a particular document is selected as follows:

$$difference = \frac{\#local\_visits}{\#local\_calls} - \frac{\#global\_visits}{\#global\_calls}$$

Difference is computed at the root of the hood for each sense of *w*. If a sense does not have a hood or if the local count at its hood root is less than 2, difference is set to 0. If a sense has multiple hoods, difference is set to the largest difference over the set of hoods. The sense corresponding to the hood root with the largest positive difference is selected as the sense of the word in the document. If no sense has a positive difference, no WordNet sense is chosen for this word.

The idea behind the disambiguation process is to select senses from the areas of the WordNet hierarchies where document-induced (local) activity is greater than the expected (global) activity. The hood construct is designed to provide a point of comparison that is broad enough to encompass markings from several different words yet narrow enough to distinguish among senses.

4.3 Text Document Classification

Assume that every word in each text document in a collection has been replaced by its senses after word sense disambiguation. In the classification phase, we actually work on senses of each word. The psuedo code is in Figure 3.

```
Procedure Classify (t: a text document)
For each sense of the words in t do
Locate the synset s in the hierarchy  //Find the synset s by searching the hierarchy
Mark s
End
Find the parents p of all the marked s   // Find the parents of all the marked synsets by
                                         following the parent-child links in the
                                         hierarchy
Return (p)
```

Fig. 3. Psuedo code of sense-based text classification

The advantages of our sense-based text classification algorithms are below:

1) The confusion incurred by ambiguation is reduced. All the keywords in a document help to determine the real sense in the context.
2) The class a given document classified into is all determined by itself, not disturbed by any user bias.
3) Since WordNet is an e-dictionary, the hierarchy of WordNet is easy to update. Therefore, the classes of all the documents are easy to update.
4) Each document may be classified in multiple classes.

5 Experimental Results

In our experiment, we use the part-of-speech tagged Brown Corpus. It consists of 1,014,312 words of running text of edited English prose printed in the United States during the calendar year 1961. This document set consists of 479 tagged documents. Each word is tagged with its certain linguistic category. It has been extensively used for natural language processing work. While the words in each grammatical class are used with a particular purpose, it can be argued that most of the semantics is carried

by noun words. Thus, nouns can be taken out through the elimination of verbs, adjectives, adverbs, connectives, articles and pronouns.

5.1 Flow of Experiment

Stemming. Stemming is a technique for reducing words to their grammatical roots. A stem is the portion of a word which is left after the removal of its affixes (i.e., prefixes and suffixes). A typical example of a stem is the word "*connect*" which is the stem for variants *connected, connecting, connection, and connections*. Stems are thought to be useful because they reduce variants of the same root word to a common concept.

Removing stopwords. Words which are too frequent among the documents are not good discriminators. In fact, a word which occurs in 80% of the documents in the document collection is useless for purpose of retrieval or classification. Such words are frequently referred to as stopwords and should be filtered out. Articles, prepositions and conjunctions are candidates for a list of stopwords, such as "an", "against", "and".

5.2 Experimental Result Analysis

We randomly choose 50 documents to classify. Since WordNet provides semantically tagged Brown Corpus files, we compare our results with the manually identified results. The accuracy rate of classification is 32%.

In order to find out the reason for the low classification accuracy, we investigate the effectiveness of word sense disambiguation of the 50 documents. Experimental results are shown in Table 1. Hit Rate is also defined.

$$Hit\_Rate = \frac{\#of\_words\_that\_are\_selected\_the\_same\_synset\_as\_manually\_identified}{\#\_of\_words\_in\_the\_stemmed\_file}$$

Table 1. Hit Rate of word sense disambiguation on Brown Corpus

Hit Rate	<15%	15%-20%	20%-25%	25%-30%	30%-35%	>40%
# docs getting the hit rate	1	7	12	16	14	0

From Table 1, we can see that the hit rate of word sense disambiguation is not as optimistic as expected. Most rates are between 15% and 35%. Therefore, the low hit rate resulted in the low classification accuracy of text documents.

Discussion

The reasons of the low hit rate of word sense disambiguation are as follows:

1) Although most of the semantics is carried by the noun words, verbs, adjectives, adverbs are important factors that can help determine appropriate senses for an ambiguous word. In order to improve the hit rate, WordNet for verbs, adjectives, adverbs cannot be ignored in further studies.

2) One word is possible to be used multiple times in one document, while each appearance may use different sense. In the current algorithm, multiple occurrences of

each word are ignored, and each word is mapped to a unique sense. Actually, an appropriate weight should be assigned to each non-stopword.

3) Part-of-speech tagger used for Brown Corpus separates words connected with underscore, such as school_board, which is an individual word while it is separated into two words "school" and "board" by the tagger. Thus, the sense of school or board will never hit the manually identified word sense of school_board. Therefore, it is also one of the factors that influence the accuracy.

6 Conclusions and Future Work

In this paper, we proposed a text classification method based on word sense disambiguation. In order to define an appropriate mid-level category for each sense, hood [5] was implemented on WordNet. Then, each non-stopword in a given document was mapped to the concept hierarchy where each synset maintains a counter. The hoods and the associated counters determined the intended sense of the ambiguous word. The nearest ancestors of the senses of all the non-stopwords were selected as the classes of a given document. We applied this algorithm to Brown Corpus. The effectiveness of our text classification method is evaluated by comparing the classification results with the classification results using manual disambiguation offered by Princeton University. We also discussed the weakness of our algorithm.

Our proposed sense-based text classification algorithm is an automatic technique to disambiguate word senses and then classify text documents. If this automatic technique can be applied in real applications, the classification of e-documents must be accelerated dramatically. It must be a great contribution to the management system of Web pages, e-books, digital libraries, etc.

In our future research, we will focus on the following aspects: 1) Build relational databases for verbs, adjectives, adverbs divisions of WordNet. Then, for each document, mapping all the non-stopwords including nouns, verbs, adjectives, adverbs to their senses. 2) Assign each non-stopword a weight which indicates its significance in a given document. For example, weight can be defined as the number of occurrence in a synset.

Acknowledgments. This work was granted in part National Natural Science Foundation of China Project #60674109, #70531040, #70472074, Ministry of Science and Technology of China 973 Project #2004CB720103.

References

1. Yang, Y., Lin, X.: A re-examination of text categorization methods. SIGIR (1999) 42-49
2. Han, E., Karypis, G.: Centroid-Based Document Classification Analysis & Experimental Result. PKDD (2000)
3. McCallum, A., Nigam, K.: A Comparison of Event Models for Naïve Bayes Text Classification. AAAI/ICML, Workshop on Learning for Text Categorization (1998)
4. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys. (2002) 34(1): 1-47

5. Voorhees, E.: Using WordNet to Disambiguate Word Senses for Text Retrieval. SIGIR (1993) 171-180
6. Wu, H., Phang, T., Liu, B., Li, X.: A Refinement Approach to Handling Model Misfit in Text Categorization. SIGKDD (2002) 207-216
7. Miller, G.: Special Issue, WordNet: An on-line lexical database. International Journal of Lexicography, 3(4) (1990)
8. Brown, P., Pietra, S., Pietra, V., Mercer, R.: Word sense disambiguation using statistical methods. Proc. of the 29th Meeting of the Association for Computational Linguistics (ACL-91), Berkley, C.A. (1991) 264-270
9. Agirre, E., Rigau, G.: Word sense disambiguation using conceptual density. Proc. of COLING (1996)
10. Richardson, R., Smeaton, A.: Using wordnet in a knowledge-based approach to information retrieval. Proc. of the BCS-IRSG Colloquium, Crewe (1995)
11. Yarowsky, D.: Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. Proc. of the 14th International Conference on Computational Linguistics (COLING-92), Nantes, France (1992) 454-460
12. Luk, A.: Statistical sense disambiguation with relatively small corpora using dictionary definitions. Proc. of the 33rd Meetings of the Association for Computational Linguistics (ACL-95), Cambridge, M.A. (1995) 181-188
13. Feldman, R., Hirsh, H.: Finding associations in collections of text. Machine Learning and Data Mining: Methods and Applications, New York: John Wiley & Sons (1998) 223-240
14. Wang, K., Zhou, S., Liew, S.: Building hierarchical classification using class proximity. Proc. of Intl. Conf. Very Large Data Bases (1999) 363-374
15. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext classification using hyper-links. Proc. of Intl. Conf. SIGMOD (1998) 307-318
16. Cowie, J., Guthrie, J., Guthrie, L.: Lexical disambiguation using simulated annealing. Proc. of COLING Conf. (1992) 359-365
17. Demetriou, GC.: Lexical disambiguation using constraint handling in Prolog (CHIP). Proc. of the European Chapter of the ACL, 6(1993) 431-436
18. Church, KW.: Using bilingual materials to develop word sense disambiguation methods. Proc. of ACM SIGIR Conference (1992) 315-350