Associating Collocations with WordNet Senses Using Hybrid Models

陳奕均 Yi-Chun Chen

國立清華大學資訊工程學系

Department of Computer Science
National Tsing Hua University

jordanchengno1@hotmail.com

顏孜羲 Tzu-Xi Yen

國立清華大學資訊工程學系

Department of Computer Science

National Tsing Hua University

joseph.yen.@gmail.com

張俊盛 Jason S. CHang

國立清華大學資訊工程學系

Department of Computer Science

National Tsing Hua University

jason.jschang@gmail.com

Abstract

In this paper, we introduce a hybrid method to associate English collocations with sense class members chosen from WordNet. Our combinational approach includes a learning-based method, a paraphrase-based method and a sense frequency ranking method. At training time, a set of collocations with their tagged senses is prepared. We use the sentence information extracted from a large corpus and cross-lingual information to train a learning-based model. At run time, the corresponding senses of an input collocation will be decided via majority voting. The three outcomes participated in voting are as follows: 1. the result from a learning-based model; 2. the result from a paraphrase-based model; 3. the result from sense frequency ranking method. The sense with most votes will be associated with the input collocation. Evaluation shows that the hybrid model achieves significant improvement when comparing with the other method described in evaluation time. Our method provides more reliable result on associating collocations with senses that can help lexicographers in compilation of collocations dictionaries and assist learners to understand collocation usages.

關鍵詞：超語意標示，搭配詞分類，詞彙語意解歧，詞網、最佳熵值模型、重述

Keywords: supersense tagging, collocation classification, word sense disambiguation, WordNet, maximum entropy model, Paraphrase.

1 Introduction

A collocation is a pair of words that co-occur with more frequency than random. A collocation usually contains a base word (e.g., "*oil*" in *fuel oil*) and a collocate (e.g., "*fuel*" in *fuel oil*). In a collocations dictionary, we can find many collocates of a base word (e.g., *fuel oil*, *motor oil*, *peanut oil, salad oil*). Some collocations dictionaries show the collocates for all senses, while other collocations dictionaries present the collocates by senses of a base word so learners can better grasp the usage of a collocation.

Determining the set of broad senses to classify collocations is not an easy task. Researches have used thesaurus topics such as Roget's (Yarowsky, 1992) or arbitrarily top-level WordNet senses as classes. There are 44 semantic classes called lexicographer-files and each synset in WordNet is assigned to one lexicographer-file. There are 26 lexicographer-files (or *supersenses*), which can be used to tag common nouns. Consider the word "*oil*" which can be *used as fuel/to make machines work smoothly, or* as belonging to the *noun.substance* supersense and *used in cooking* could be seen as belonging to the *noun.food* supersense.

In this paper, we present a hybrid model that automatically associated a given collocation with the corresponding supersense. The hybrid model is composed of a learning-based method, a paraphrase-based method and a sense frequency ranking method. The output supersense of a collocation is decided via majority vote of the above three methods.

At training time, we need some collocations tagged with supersenses as seeds. There are a huge number of collocations in WordNet, so we can use those collocation and supersense pairs to train the model. Sentences containing the input collocations extracted from a large corpus and Chinese translation of the collocations are used as features of the model. We will descript the training process in more details in Chapter 3.

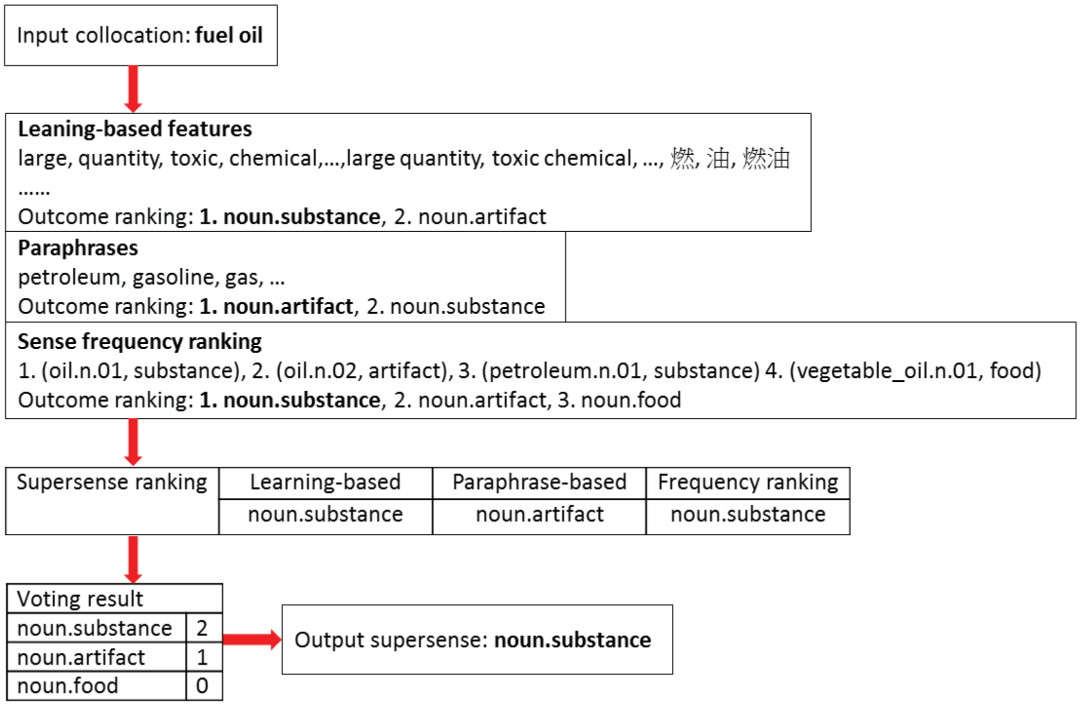
Input collocation: fuel oil
Leaning-based features
large, quantity, toxic, chemical,…,large quantity, toxic chemical, …, 燃, 油, 燃油
……
Outcome ranking: 1. noun.substance, 2. noun.artifact
Paraphrases
petroleum, gasoline, gas, …
Outcome ranking: 1. noun.artifact, 2. noun.substance
Sense frequency ranking
1. (oil.n.01, substance), 2. (oil.n.02, artifact), 3. (petroleum.n.01, substance) 4. (vegetable_oil.n.01, food)
Outcome ranking: 1. noun.substance, 2. noun.artifact, 3. noun.food
Supersense ranking
Learning-based
noun.substance
Paraphrase-based
noun.artifact
Frequency ranking
noun.substance
Voting result
noun.substance 2
noun.artifact 1
noun.food 0
Output supersense: noun.substance

Figure 1. An example procedure for associating the collocation of fuel oil with a supersense *noun.substance*

An example procedure for associating the collocation of *fuel oil* with a supersense *noun.substance* is shown in Figure 1. We extract sentences containing the input collocation from a corpus and take the sentences and Chinese translation as features. Then, we use the pre-trained machine learning model to predict the supersense. Second, we use the words similarity and words dependency relations to paraphrase the base word. Then, we calculate the WordNet similarity of base word and the paraphrases to identify the supersense. Third, we simply list the lexicographer-files of the input collocation base word and choose the first one as the supersense since the order of the list corresponds to the sense frequency of that word. At last, a relative majority vote for the three results determines the final output.

The experimental results show that our hybrid method can automatically associate collocations with supersenses with a higher performance than the baseline method. The results can also be used to help lexicographers in compilation of a collocations dictionary. Furthermore, learners could understand the usage of collocations in a specific sense.

2 Related Work

Associating collocations with supersenses in WordNet is similar to Word Sense Disambiguation (WSD), the process of identifying the meaning of a specific word in a given context. In this paper, we address a special case of disambiguating the headword of a given collocation.

Previous work in WSD is mostly based on some kind of machine learning models. Hearst (1991) uses a set of orthographic, syntactic and lexical features to train large text corpora and disambiguates noun homographs. Yarowsky (1992) uses Naïve Bayesian model to train large corpora to disambiguate words to Roget’s Thesaurus categories. Leacock, Towell and Voorhees (1993) bases on Bayesian decision theory, neural networks and content vectors to train the knowledge about patterns of words co-occurrences and disambiguates words to WordNet senses. The main disadvantage is that the demand of annotated training data which are time-consuming and labor intensive to obtain.

In a work more closely to our research, Inumella, Kilgarriff and Kovar (2009) try to assign the collocations for a word that automatically identified from a large corpus to its distinct senses. Their short term goal is to generate a new English collocations dictionary (Macmillan Collocation Dictionary). Most of the previous works focus on words level, while this research focuses on collocations. We describe two of their automatic approaches: Thesaurus method and Yarowsky's method (1995). The thesaurus method works on the promise that a sense shares its collocates with its thesaurus class members. For example, consider a thesaurus class with six members {cricket, butterfly, leech, worm, bee, queen}, they extract collocates such as young, fly, feed, breed that at least appear in two class members and insert them to that sense. Another method is Yarowsky's method, which relies on the heuristic of "one sense per collocation" (Yarowsky, 1993) and "one sense per discourse" (Gale, Church and Yarowsky, 1992). The algorithm first collects some seed collocations with senses by dictionary-parsing and uses supervised classification algorithms for training and labeling. Then they add new labeled collocations to training set and repeat labeling. Finally, they use a decision list algorithm to terminate.

In contrast to previous works in Word Sense Disambiguation and semantic classification, we present a hybrid system that automatically associates collocations to supersenses using a learning-based method, a paraphrase-based method and a sense frequency ranking method, with the goal to help lexicographers in compilation of collocation dictionaries and help learners to better grasp the usage of a collocation. We describe the method in more details in the next chapter.

3 Method

Associating collocations (e.g., required course) with dictionary senses often does not work very well. To obtain a better performance, we introduce a learning-based method using context and cross-lingual features, a paraphrase-based method using words similarity relation and dependency relation, and a sense frequency ranking method.

3.1 Problem Statement

We focus on automatically associating collocations with corresponding supersenses. The output senses could be used by lexicographers to save effort in compile collocations dictionaries and learners can better grasp the usage of a collocation. Supersenses are 26 lexicographer-files in WordNet noun hierarchy chosen by lexicographers and are believed to be general enough for sense allocation.

3.2 Training Sense Assignment Models

In this section, we explain our approaches to find the supersense including a learning-based method, a paraphrase-based method and a sense frequency ranking method. Figure 2 describes the processes of our methods.

(1) Generate collocation and supersense pairs from WordNet (Section 3.2.1)

(2) Train machine learning model from corpus for collocations (Section 3.2.2)

(3) Obtain supersense using machine learning model (Section 3.2.3)

(4) Obtain supersense using similarity and dependency information (Section 3.2.4)

(5) Obtain supersense using sense frequency ranking from WordNet (Section 3.2.5)

Figure 2. Outline of the process for obtaining supersense in different approaches

fuel oil	noun.substance
electrical discharge	noun.event
busy day	noun.time
required course	noun.act
fitted sheet	noun.artifact
bus driver	noun.person

Figure 3. Example of collocation and supersense pairs extracted from WordNet

3.2.1 Generating Collocation and Supersense Pairs

In the first stage (Step (1) in Figure 2), we attempt to find a set of collocations and their pre-tagged supersenses pairs

$$CSS = (< Col_1, S_1 >, < Col_2, S_2 >, < Col_3, S_3 >, \dots, < Col_i, S_i >)$$

as seeds collocations to train a machine learning model M from WordNet. For example, the supersense for a collocation *fuel oil* is *noun.substance*. Examples of collocation and supersense pairs extracted from WordNet are shown in Figure 3.

We use two heuristics to achieve this goal. First, we go through each hyponyms of noun synsets and examine their lemma names to find collocations. For example, consider a synset *Synset('discharge.n.01')*, one of its lemma name is *discharge* and one of its hyponyms is *Synset('electrical_discharge.n.01')* with a lemma name *electrical_discharge*. Since the base word of *electrical_discharge* matches *Synset('discharge.n.01')*'s lemma name *discharge*, we can take *electrical discharge* as a collocation and the lexicographer-file of Synset('discharge.n.01') *noun.event* as a supersense to form the $< collocation, supersense >$ pair, (*electronic discharge, noun.event*).

Second, we search the collocations from definitions D_{wn} and example sentences E_{wn} of each noun synset. We utilize a parser to generate part-of-speech and lemma for D_{wn} and

E_{wn}. For a noun synset syn_i, one of its lemma name is lem_i, and the definition or example i as one of our selected $< collocation, supersense >$ pair. For example, a synset *Synset('day.n.05')* has one lemma name ***day*** and one example sentence "*it was a **busy day** on the stock exchange*". So we can take *busy day* as a collocation and the lexicographer-file *noun.time* as a supersense to form the $< collocation, supersense >$ pair, (*busy day, noun.time*).

3.2.2 Training Machine Learning Model

In the second stage (Step (2) in Figure 2), we use the collocation and supersense pairs obtained in section 3.2.1 to find sentences to train a sense classifier. First, a parser is used for generating part-of-speech tag and lemma form for all sentences in the monolingual corpus p and search from on-line machine translating system MT for Chinese collocation translation.

For example, consider the collocation required course and its supersense noun.act. We can find sentence such as "A required course for all students, to be completed before the end of the third year, and to be examined by individual colleges" from MC and its Chinese collocation translation "必修課" from on-line translation resource. The base word course has 6 different supersenses, but the words like students, third year, examined, colleges are highly related to the collocation required course and the supersense noun.act rather than other supersenses such as noun.food, noun.artifact or noun.object. The Chinese translation provides cross-lingual information like "課" to disambiguate the sense of course. The other translation for course like "路線" or "餐" would lead to different supersenses.

The input to this stage is a set of features. The above example *required course* showed that context words of a collocation may contain some words highly related to the corresponding supersense and cross-lingual information for a collocation also helps to disambiguate the supersense. So the features we use for one training event are

(1) unigram and bigram of a sentence extracted from MC_p containing the collocation

(2) Chinese translation of the collocation from MT

For each $< Col, S >$ pairs in CSS, we extract sentences containing Col from MC_p as $Sentences$ and obtain Chinese translation of Col from MT as $Trans$. Then, for each sentence $Sent$ in $Sentences$, we extract unigram Uni and bigram Bi from $Sent$. Note that stopwords are filtered for both Uni and Bi. The next step, we use Uni, Bi and $Trans$ as features and S as the standard output supersense to append machine learning event to $Features$. Note that $Trans$ actually transforms to a list of unigram and bigram of Chinese words while training. The output of this stage is a probability model M trained from a set of training events $Features$ for predicting the collocation supersenses using a machine learning tool ML.

3.2.3 Obtaining supersense using machine learning model

In the third stage (Step (3) in Figure 2), we attempt to predict the supersense for the input collocation (C, B) using the machine learning model M described in Section 3.2.2. The runtime procedure is similar to the training algorithm.

First, we extract sentences containing (C, B) from MC_p as $Sentences$ (Step(1a) in Figure 4) and use on-line machine translation system MT to obtain Chinese translation of (C, B) as $Tran$ (Step(1b)). For associating (C, B) with a supersense, we only consider i in $Sentences$, we extract unigram Uni (Step (2a)) and bigram Bi (Step (2b)) from $Sent$. Stopwords are filtered for both Uni and Bi similar to what is done at training time. Then, Uni, Bi and $Tran$ are combined together to predict the supersenses using M. The output of M is a supersense probability list $predictList$ that contains all supersenses and the probability for (C, B) (Step 3)).

Algorithm 2. Obtaining supersense using machine learning model

```
PROCEDURE MachineLearningEvaluateSupersense($(C, B)$)
(1a)  $Sentences$ = extractSentences($(C, B)$,$MC_p$)
(1b)  $Trans$ = getTranslation($(C, B)$)
(1c)  $Candidates$ = getLexFiles($B$)
      $topScore = \emptyset$, $topSense = \emptyset$, $freq = \emptyset$, $totalProb = \emptyset$, $avgProb = \emptyset$, $outcome = \emptyset$
      for each $Sent_i$ in $Sentences$
(2a)      $Uni$ = extractUnigram($Sent_i$)
(2b)      $Bi$ = extractBigram($Sent_i$)
(3)       $predictList = M(Uni, Bi, Trans)$
          for each ($sense_j$, $prob_j$) in $predictList$
              if $sense_j$ in $Candidate$ and ($prob_j > numProb$)
(4a)              $tmpScore_j = prob_j$
(4b)              $tmpSense_j = sense_j$

(5a)      $topScore_i$ = Max($tmpScore_j$)

(5b)      $topSense_i = tmpSense_j$ that has Max($tmpScore_j$)

(5c)      $totalProb[topSense_i]$ += $topScore_i$
(5d)      $freq[topSense_i]$ += 1
      for each $sense$ in $totalProb$
(6)       $outcome[sense]$ = ($freq[sense]$, $totalProb[sense]/freq[sense]$)
(7)   $rankedSenses$ = Sort $outcome$ in decreasing order of $freq$, if more than 1 $sense$
                       share same frequency, sofrt those sense in decreasing order of
                       average probability
(8)   Return the top $rankedSenses$
```

Figure 4. Algorithm for obtaining supersense using machine learning model

We go through each $(sense_j, prob_j)$ in $predictList$ and keep $(sense_j, prob_j)$ as $(tmpSense_j, tmpScore_j)$ if both $sense_j$ in $Candidate$ and $prob_j$ higher than a probability threshold $numProb$ (Step (4a and 4b)). Then we choose maximum $tmpScore_j$ as $topScore_j$, the corresponding $tmpSense_j$ as $topSense_j$ (Step (5a) and (5b)). A dictionary $totalProb$ is used to store the probability sum $topScore_j$ of each distinct $topSense_j$, and another dictionary $freq$ is used to store the frequency of each distinct $topSense_j$ (Step (5c) and (5d)). For each $sense$ in $totalProb$, we store the frequency $freq[sense]$ and the average sense probability $totalProb\, sense\ /freq[sense]$ to $outcome$ (Step (6)). Then, we sort $outcome$ in decreasing order of frequency as $rankedSenses$. If there is more than one sense in $outcome$ that have the same frequency, they would be sorted in decreasing order of the average sense probability. Finally, we output LB (Step (7)).

Corpus-based machine learning for associating collocations with supersenses can reduce the sense dominance problem, since context words of different supersenses are generally different and translations of a same base word in different senses tend to be different, too. With this in mind, we use sentences of a collocation extracted from a corpus and the collocation translation to disambiguate the supersenses of the base word of a given collocation.

3.2.4 Obtain supersense using similarity & dependency information

In the fourth stage (Step (4) in Figure 2), we use a paraphrase-based strategy to determine the supersense. A paraphrase is a restatement of the meaning of a text or passage using another form. By calculating the similarity between a collocation and its paraphrases, we can determine its supersense. This method is based on the assumption that original collocation shares the same supersense with its paraphrases.

For example, consider an input collocation *fitted sheet* using the paraphrase method. The word *sheet* has four supersenses: *noun.object, noun.communication, noun.artifacrt, noun.shape* in WordNet. Paraphrase candidates of *fitted sheet* are *coat, cloth, plate, pan, foil, plastic* identified base on similar words list of *sheet* and *coat,*

Table 1. Example similar words and dependent words of *required course*

Similar words of *sheet*	Dependency relation of *fitted*

('plate', 0.16), ('sheeting', 0.15)	('jacket', 9), ('suit', 5)
('pan', 0.14), ('steel', 0.14)	('bodice', 3), ('less', 3)
('coat', 0.13), ('tube', 0.12)	('gown', 2), ('Top', 2)
('metal', 0.12), ('paper', 0.12)	('carpet', 1), **('cloth', 1)**
('slab', 0.11), ('pipe', 0.11)	**('coat', 1)**, ('leader', 1)
('layer', 0.11), ('cold-rolled', 0.11)	('plaid', 1), ('topper', 1)
('stainless', 0.11), ('surface', 0.11)	('version', 1), ('a little', 1)
('glass', 0.11), ('tubing', 0.11)	('long', 1), ('uniquely', 1)
('booklet', 0.11), ('cut-sheet', 0.11)	('around', 1), ('than', 1)
('cloth', 0.11), …	

cloth, jacket, suit based on dependency relations list of *fitted*. The intersection of the two candidate list contains *coat* and *cloth*. It means that *coat* and *cloth* are paraphrases of *sheet* when collocating with *fitted*. The example similar words and dependency relations of *fitted sheet* is shown in Table 1.

Subsequently, we compare the synsets similarity for both *(coat, sheet)* and *(cloth, sheet)*. The top-ranked similarity of *(coat, sheet)* is *((Synset('coating.n.01'), Synset('sheet.n.06')), 0.769)* and the lexicographer-file of *Synset('sheet.n.06')* is *noun.artifact*; the top-ranked similarity of *(cloth, sheet)* is *((Synset('fabric.n.01'), Synset('sail.n.01')), 0.857)* and the lexicographer-file of *Synset('sail.n.01')* is *noun.artifact*. So the frequency of *noun.artifact* is 2, while other supersenses are all 0. We then output *noun.artifact* as the supersense of input collocation *fitted sheet*.

By using paraphrase-based method, words that related to the input collocation can be the extracted. The collocation could be disambiguated since most of the words with other senses tend not to share the paraphrases. So we can find the sense relation between input collocation and extracted words to obtain the supersense.

3.2.5 Obtaining supersense using sense frequency ranking

In the last stage (Step (5) in Figure 2), we use the sense frequency to identify the supersense. In many previous works on WSD, sense frequency plays an important role to indicate the sense. A word may have different senses, but most of time, it tends to associated with the dominant sense. So for disambiguating word senses, choosing the most frequent sense is often used as a baseline.

Many sense frequency methods are based on sense estimation in a corpus. But here we use the sense frequency information in WordNet. For any word in WordNet, there are one or more synsets and the synsets are listed in decreasing order of frequency. So we can simply return the first synset as the supersense. Sense frequency ranking method has the highest coverage, and that is important since our goal is to disambiguate all collocations. We also use this method as the baseline method to compare with our results. We will describe the details of evaluation in Chapter 4.

3.2.6 The Runtime Hybrid Process

Once the learning-based procedure, the paraphrase-based procedure and the sense frequency ranking procedure produce the supersenses, a relative majority vote is carried out to FR are the three predicted supersense described in sections 3.2.2 to 3.2.4. Each supersense has one vote and the supersense with the most votes is the final output S. As shown in Figure 2, after running the three procedures for collocation *fuel oil*, we obtain *noun.substance*, *noun.artifact* and *noun.substance*. The supersense *noun.substance* has 2 votes and *noun.artifact* has 1, so the final output S is *noun.substance*.

Sometimes the three procedures produce 3 different supersenses without an agreement. Moreover, the learning-based procedure or the paraphrase-based procedure produce no results, because either the sentences containing the input collocation cannot be found in corpus MC, or the paraphrases of the input collocation cannot be found and leads the voting has no agreement. In this case, we use back-off to find the supersense. When there is no agreement FR. As long as the base word of the input collocation exists in WordNet, we can produce an output.

4 Experimental Setting

We have proposed a hybrid model to associate collocations with broad sense classes, with the goal of helping lexicographers in compilation of collocation dictionaries. The evaluation focuses on the intended supersenses of a set of collocations produced by the proposed system. We extracted a set of collocation and supersense pairs from WordNet, so the evaluation could be done automatically.

4.1 Data set

In our experiment, we used WordNet, a large lexical database of English which contains approximately 117,000 synsets and 155,000 sense-disambiguated words and collocations, to generate the collocations for training, developing and testing. As we have described in Section 3.2.1, collocations are extracted from WordNet using two heuristics:

(1) extract collocations from hyponyms of noun synsets

(2) extract collocations from definitions and examples sentences of noun synsets

We extracted 18,586 collocation and supersense pairs from (1) and 1,784 pairs from (2). The extracted collocations were filtered through a collocation list. The collocation we used is a list of base word/collocates pairs for the top 60,000 lemmas from the Corpus of Contemporary American English (COCA) (Davies, 2008) which contains 4,200,000 collocations. After this step, the total number of collocations was reduced to 7,489. With heuristic (2), we used GENIA tagger (Tsuruoka, 2005) which analyzes English sentences and outputs the base forms, part-of-speech tags, chunk tags, and named entity tags to tag the definitions and example sentences.

We randomly selected 829 collocations as development set and 6,660 for training and testing from the collocation and supersense pairs. For training and testing, we split the 6,660 collocations into 10 parts that each part contains 666 collocations and we ran ten-fold validation to evaluate the performance of each part.

In learning-based procedure, we employed *Maximum Entropy* (*ME*) model to associate input collocations with supersenses. *ME* is a flexible statistical learning model that aims to maximize the entropy when characterizing some unknown events. The model estimates outcomes according to a set of features with least possible bias. The *ME* model we used for training and testing is Maximum Entropy Modeling Toolkit for Python and C++ (Zhang, 2004). The features we used for the *ME* model is extracted from *British National Corpus* (*BNC*), a 100 million word collection of samples of written and spoken language from a wide range of sources. We use *GENIA tagger* to tag the sentences in *BNC* and filtered the stopwords in the sentences using *Natural Language Tool Kit* (*NLTK*), a suite of open source program modules written in Python (Loper and Bird, 2002). More specifically, we used the *stopwords* in *nltk.corpus* and obtained the English stopwords list. Another feature, the Chinese translation of the collocations, was obtained from *Google Translate*.

In the paraphrase-based procedure, we use a set of words with similar words which contains 100,000 words and about 24,000,000 similar words and words with dependency relations which contains 20,000,000 dependency relations. The data is obtained using *MINIPAR* (Lin, 1993), a broad-coverage parser for the English language. The similarity comparison algorithm for words used in this stage is JCN similarity (Jiang and Conrath, 1997). JCN similarity bases on the information content (IC, a measure of the specific of a concept) of the Least Common Subsumer (LCS, most specific ancestor node). According to (Sinha and Mihalcea, 2007), JCN similarity tends to work best for nouns.

4.2 Methods compared

Our approach starts with an adjective-noun or noun-noun collocation given by a user, and determines the corresponding sense to the input collocation using external resources related to the input collocation. The output of our system is a supersense in WordNet associated with the input collocation that can be used to help lexicographers in compiling collocation dictionaries, or shown to English learners directly.

In this paper, we have proposed a hybrid model for associating collocations with supersenses, in which we used a learning-based model, a paraphrase-based similarity comparison, and a sense frequency ranking method. Therefore, we compare the results based on each method and combination of the above methods for evaluating the system performance in more details.

We compare different methods to associating the collocation with supersense using the test set described in Section 4.1. The methods evaluated for the comparison are listed as follows:

— **FR**: Sense frequency ranking method as we described in Section 3.2.5, using the sense frequency information to determine the supersense of a collocation. This method is also the baseline method in our experiment.

— **LB**: Learning-based method as we described in Section 3.2.3, using learning-based method to determine the supersense of a collocation.

— **LB+FR**: Combinational method of learning-based method and sense frequency ranking method, using **FR** as a back-off if **LB** cannot be applied.

— **PB**: Paraphrase-based method as we described in Section 3.2.4, using similarity and dependency relations of a collocation to determine the sense of that collocation.

— **PB+FR**: Combinational method of paraphrase-based method and sense frequency ranking method, using **FR** as a back-off if **PB** cannot be applied.

— **LB+PB**: Combinational method of learning-based method and paraphrase-based method, using **PB** as a back-off if **LB** cannot be applied.

— **LB+PB+FR**: Hybrid method of all methods we proposed. The running sequence is **LB→PB→FR** that **LB** determines all the test set, then **PB** determines those **LB** cannot solve, then **FR** determines those **PB** cannot solve.

— **MV+BO**: The most complete version of the system we proposed. First, we run the test set using all three methods **LB**, **PB** and **FR** and use relative majority vote to rank supersense results. The rest of collocations that cannot be determined run in the following sequence **LB→PB→FR**.

5 Evaluation Result and Discussion

In this chapter, we report the evaluation results of our experiments using methodologies and the settings we described in Chapter 4. We evaluated 8 different methods as described in Section 4.2. We ran ten-fold validation on 6,660 random selected collocations. We report the average performance of the 10 test results. For non-learning based method, we evaluated the whole 6,660 collocations. Table 2 shows the performance for development dataset and test dataset in 8 different methods based on *precision*, *recall* and *F-measure*.

Table 2. Performance for development dataset and test dataset in 8 different methods based on *precision*, *recall* and *F-measure*

strategy	Development Set			Test Set		
	Prec.	Rec.	F-m.	Prec.	Rec.	F-m.
FR (baseline)	.74	.74	.74	.75	.75	.75
LB	.80	.61	.69	.80	.62	.70
LB+FR	.78	.78	.78	.80	.80	.80
PB	.79	.57	.66	.76	.55	.63
PB+FR	.78	.78	.78	.76	.76	.76
LB+PB	.80	.72	.76	.80	.72	.75
LB+PB+FR	.80	.80	.80	.80	.80	.80
MV+BO	.81	.81	.81	**.81**	**.81**	**.81**

For comparison, we used the baseline of sense frequency ranking method **FR** with 75% precision, recall and F-measure. The learning-based method **LB** achieves the precision 80% and recall 62% with 5% increases in precision. But the recall decreases since no sentences containing the collocations are found in the corpus. Those collocations are not given a supersense. If we add **FR** to the system as **LB+FR**, the precision, recall and F-measure increases to 80%. The paraphrase-based method **PB** on development dataset has a 5% increase on precision comparing with baseline, but on test dataset, the precision decreases to 76% with a low recall of 55%. The low recall is due to the fact that many collocations paraphrases cannot be found. For this we also add **FR** to the system as a back-off and the precision, recall and F-measure of **PB+FR** increases to 76%. The experimental result on **LB+PB** shows that the precision maintains on 80%, and recall increases nearly 10% comparing with **LB** and achieves the highest recall in all the methods without **FR**.

The performance of **LB+PB+FR** reaches 80%, the same as **LB+FR** since the performance of **PB** is not good as **LB**. We believe that using a relative majority vote to determine the supersense would lead a better performance. **MV+BO** confirms our hypothesis and achieves the best performance of precision, recall and F-measure 81%. The precision of majority vote that has 3 votes is 95% with recall of 33% while the majority with 2 votes is precision 79% and recall 34%. So with more than 2 votes, the precision reaches 87% with a recall of 67% and F-measure of 76%.

Take a deeper look in the sense dominance problem we mentioned in Chapter 3. Previous work suffered from that the collocations are often associated with dominant senses. We show the performance of **MV+BO** when dealing with two different condition: (1) most frequent sense collocations, (2) non-most frequent sense collocations. We could see that when dealing with most frequent sense collocations, 93% of collocations can be correctly associated with supersenses. When dealing with non-most frequent sense collocations, we are still able to correctly associate 46% of collocations with supersenses. So we prove that the sense dominance problem can be reduced by using our hybrid algorithm.

6 Conclusion

Many avenues exist for future research and improvement of our system. For example, in the learning-based method, the recall could increase by using a larger corpus or the web data to extract more sentences as collocations' features. The cases where the sentences of the input collocation are not found in a corpus could be reduced. Additionally, we could improve the quality of collocation translations to improve the performance of the learning-based method. In the paraphrase-based method, both precision and recall are not satisfactory, but we still believe that the method has potential. By generating a new similar words list and dependency relations list using a large corpus could produce better paraphrases for associating collocations with supersenses and increasing the recall. Most of the 26 supersenses are natural and reasonable. However, we still find that some supersenses are not very intuitive and may cause problems in tagging. So finding more appropriate set of classes is worth further study.

In summary, we have introduced a hybrid method to automatically associate collocations with supersenses. Our goal is to help lexicographers in compilation of a collocation dictionary and help learners to better grasp the usage of a collocation. Our method is composed of a learning-based model, a paraphrase-based method, and a sense frequency ranking method. In our evaluation, we have shown that the hybrid method is significantly better compared with other methods described in this paper. And we also prove that our model can partially reduce the sense dominance problem.

References

[1] Baker L. D. and McCallum A. K., "Distributional clustering of words for text classification." Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval ACM, p. 96 , 1998.

[2] Curran J. R., "Supersense tagging of unknown nouns using semantic similarity." in proceedings of the 43rd annual meeting on association for computational linguistics Association for Computational Linguistics, 26 p., 2005

[3] Davies M. 2008. The corpus of contemporary american english (coca): 400 million

words, 1990-present. Available Online at http://www.Americancorpus.Org .

[4] Fellbaum C. "WordNet" in Theory and Applications of Ontology: Computer Applications, pp. 231-43, 2010.

[5] Gale W. A., Church K. W. and Yarowsky D., "One sense per discourse," in proceedings of the workshop on speech and natural language Association for Computational Linguistics, p. 233, 1992.

[6] Inumella A, Kilgarriff A, Kovar. Associating collocations with dictionary senses.

[7] Jiang JJ and Conrath DW., Semantic similarity based on corpus statistics and lexical taxonomy. Arxiv Preprint Cmp-lg/9709008, 1997

[8] Leacock C., Towell G. and Voorhees E., "Corpus-based statistical sense resolution." Proceedings of the ARPA workshop on human language technology. p. 260, 1993.

[9] Lin D., Dependency-based evaluation of MINIPAR. Treebanks, pp. 317-29, 2003.

[10] Miller GA. WordNet: A lexical database for English. Commun ACM 38(11):39-41., 1995.

[11] Pearce D., "Synonymy in collocation extraction." Proceedings of the workshop on WordNet and other lexical resources, second meeting of the north american chapter of the association for computational linguistics. 41 p., 2001.

[12] Tsuruoka Y, Tateishi Y, Kim JD, Ohta T, McNaught J, Ananiadou S, Tsujii J., "Developing a robust part-of-speech tagger for biomedical text." In Advances in Informatics, pp. 382-92, 2005.

[13] Yarowsky D. "Unsupervised word sense disambiguation rivaling supervised methods." Proceedings of the 33rd annual meeting on association for computational linguistics Association for Computational Linguistics. 189 p., 1995.

[14] Yarowsky D. "Word-sense disambiguation using statistical models of roget's categories trained on large corpora." Proceedings of the 14th conference on computational linguistics-volume 2 Association for Computational Linguistics. 454 p., 1992.