Computational Linguistics and Chinese Language Processing
Vol. 10, No. 3, September 2005, pp. 303-328

Using Lexical Constraints to Enhance the Quality of Computer-Generated Multiple-Choice Cloze Items

Chao-Lin Liu*, Chun-Hung Wang* and Zhao-Ming Gao+

Abstract

Multiple-choice cloze items constitute a prominent tool for assessing students' competency in using the vocabulary of a language correctly. Without a proper estimation of students' competency in using vocabulary, it will be hard for a computer-assisted language learning system to provide course material tailored to each individual student's needs. Computer-assisted item generation allows the creation of large-scale item pools and further supports Web-based learning and assessment. With the abundant text resources available on the Web, one can create cloze items that cover a wide range of topics, thereby achieving usability, diversity and security of the item pool. One can apply keyword-based techniques like concordancing that extract sentences from the Web, and retain those sentences that contain the desired keyword to produce cloze items. However, such techniques fail to consider the fact that many words in natural languages are polysemous so that the recommended sentences typically include a non-negligible number of irrelevant sentences. In addition, a substantial amount of labor is required to look for those sentences in which the word to be tested really carries the sense of interest. We propose a novel word sense disambiguation-based method for locating sentences in which designated words carry specific senses, and apply generalized collocation-based methods to select distractors that are needed for multiple-choice cloze items. Experimental results indicated that our system was able to produce a usable cloze item for every 1.6 items it returned.

Keywords: Computer-assisted language learning, Computer-assisted item generation, Advanced authoring systems, Natural language processing, Word sense disambiguation, Collocations, Selectional preferences

* Department of Computer Science, National Chengchi University, Taipei 11605, Taiwan
E-mail: chaolin@nccu.edu.tw （劉昭麟及王俊弘，臺北市文山區，國立政治大學資訊科學系）
+ Department of Foreign Languages and Literatures, National Taiwan University, Taipei 10617, Taiwan
E-mail: zmgao@ntu.edu.tw （高照明，臺北市大安區，國立臺灣大學外國語文學系）

[Received February 3, 2005; Revised June 15, 2005; Accepted June 17, 2005]

1. Introduction

Due to the advent of modern computers and the Web, academic research on intelligent tutoring systems (ITSs) have grown in the last decade. Figure 1 shows a possible functional structure of the main components of an ITS that uses test items to assess students' competence levels. With the development of mature techniques for intelligent systems and the abundant information now available on the Internet, a computer-assisted *Authoring Component* that can help course designers construct large databases of high-quality test items and course materials has become possible [Irvine and Kyllonen 2002; Wang *et al.* 2003]. With *Test-Item* and *Course-Material Databases*, the *Tutoring Component* must find ways to provide materials appropriate for students. In the ideal case, we should be able to determine students' competence levels effectively and efficiently by means of various forms of assessment and provide course materials that are tailored to each individual student's particular needs [van der Linden and Hambleton 1997; van der Linden and Glas 2000; Liu 2005]. For this purpose, we need to have appropriate techniques and a *Student-Model Database* that together enable the *Adaptive Tester* and *Course Sequencer* to identify students' competence levels, predict their needs, and provide useful course materials. When the tutoring component cannot meet students' needs, the students should be able to feedback their requests or complaints to the course designers to facilitate future improvements.

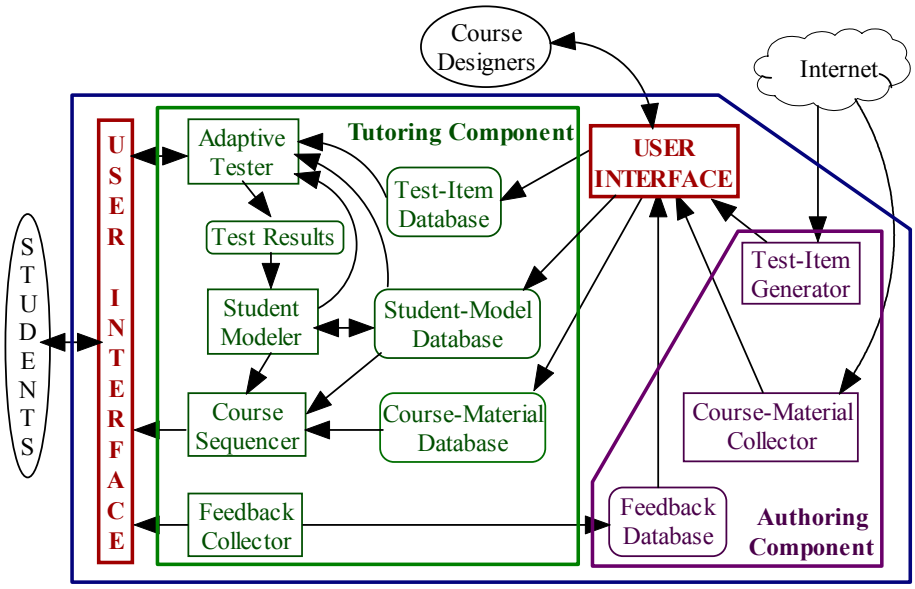

Figure 1. A functional structure of an intelligent tutoring system

As shown in Figure 1, the quality and quantity of test items are crucial to the success of the whole system, as the decisions for adaptive interactions with students depend heavily on students' responses to test items. Good test items help teachers identify students' competence levels more efficiently, and a large quantity of test items avoids the overuse of particular test items, thereby increasing the security of the item database [Dean and Sheehan 2003; Oranje 2003]. Although human experts can create test items of very high quality, the costs involved in using human experts exclusively in the authoring task can be formidable. It is thus not surprising that computer-assisted item generation (CAIG) has attracted the attention of educators and learners, who find that it offers several desirable features of generated item pools [Irvine and Kyllonen 2002]. CAIG offers the possibility of creating a large number of diverse items for assessing students' competence levels at relatively low cost, while alleviating problems related to keeping the test items secure [Dean and Sheehan 2003; Oranje 2003].

In this paper, we concern ourselves with a fundamental challenge for computer assisted language learning (CALL) and propose tools for assembling multiple-choice cloze items that are useful for assessing students' competency in the use of English vocabulary. If it is unable to determine ability to understand vocabulary, an ITS cannot choose appropriate materials for such CALL tasks as reading comprehension. To demonstrate our main ideas, we tackle the problem of generating cloze items for the college entrance examinations in Taiwan [Taiwan CEEC 2004]. (For the sake of brevity, we will henceforth use *cloze items* or *items* instead of *multiple-choice cloze items* when there is no obvious risk of confusion.) With the growth of the Web, we can search and sift online text sources for candidate sentences and come up with a list of cloze items economically with the help of natural language processing techniques [Gao and Liu 2003; Kornai and Sundheim 2003].

Techniques for natural language processing can be used to generate cloze items in different ways. One can create sentences from scratch by applying template-based methods [Dennis *et al.* 2002] or more complex methods based on some predetermined principles [Deane and Sheehan 2003]. One can also take existing sentences from a corpus and select those that meet the criteria for test items. Generating sentences from scratch provides a basis of obtaining specific and potentially well-controlled test items at the costs of more complex systems, e.g., [Sheehan *et al.* 2003]. On the other hand, since the Web puts ample text sources at our disposal, we can also filter texts to obtain candidate test items of higher quality. Administrators can then select really usable items from these candidates at relatively low cost.

Some researchers have already applied natural language processing techniques to compose cloze items. Stevens [1991] employed the concepts of concordancing and collocation to generate items using general corpora. Coniam [1997] applied factors such as word frequency in a tagged corpus to create test items of particular types. In previous works, we

considered both the frequencies and selectional preferences of words when utilizing the Web as the major source of sentences for creating cloze items [Gao and Liu 2003; Wang *et al.* 2003].

Despite the recent progress, more advanced natural language processing techniques have not yet been applied to generate cloze items [Kornai and Sundheim 2003]. For instance, many words in English carry multiple senses, and test administrators usually want to test a particular usage of a word. In this case, blindly applying a keyword matching method, such as a concordancer, may result in a long list of irrelevant sentences that will require a lot of postprocessing work. In addition, composing a cloze item requires more than just a useful sentence. Figure 2 shows a sample multiple-choice item, where we call the sentence with a gap the **stem**, the answer to the gap the **key**, and the other choices the **distractors** of the item. Given a sentence for a particular key, we still need distractors for a multiple-choice item. The selection of distractors affects the *item facility* and *item discrimination* of a cloze item and is a vital task [Poel and Weatherly 1997]. Therefore, the selection of distractors also calls for more deliberate strategies, and simple considerations alone, such as word frequency [Gao and Liu 2003; Coniam 1997], may not result in high-quality multiple-choice cloze items.

1. My sister is _____, that is, I am going to be an uncle soon.
   (A) supposing (B) assigning
   (C) expecting (D) scheduling

Figure 2. A multiple-choice cloze item for English

To remedy these shortcomings, we propose a novel integration of dictionary-based and unsupervised techniques for word sense disambiguation for use in choosing sentences in which the keys carry the senses chosen by test administrators. Our method also utilizes the techniques for computing collocations and selectional preferences [Manning and Schütze 1999] for filtering candidate distractors. Although we can find many works on word sense disambiguation in the literature [Edmonds *et al.* 2002], providing a complete overview on this field is not the main purpose of this paper. Manning and Schütze [1999] categorized different approaches into three categories: supervised, dictionary-based, and unsupervised methods. Supervised methods typically provide better chances of pinpointing the senses of polysemous words, but the cost of preparing training corpora of acceptable quality can be very high. In contrast, unsupervised methods can be more economical but might not produce high-quality cloze items for CALL applications. Our approach differs from previous dictionary-based methods in that we employ sample sentences of different senses in the lexicon as well as the definitions of polysemous words. We compare the definitions of the competing senses of the key based on a generalized notion of selectional preference. We also compare the similarities

between the candidate sentence, which may become a cloze item, and samples sentences which contain the competing senses of the key. Hence, our approach is a hybrid of dictionary-based and unsupervised approaches. Results of empirical evaluation show that our method can identify correct senses of polysemous words with reasonable accuracy and create items of satisfactory quality. In fact, we have actually used the generated cloze items in freshmen-level English classes at National Chengchi University.

We analyze the cloze items used in the college entrance examinations in Taiwan, and provide an overview of the software tools used to prepare our text corpus in Section 2. Then, we outline the flow of the item generation process in Section 3. In Section 4, we elaborate on the application of word sense disambiguation to select sentences for cloze items, and in Section 5, we delve into the application of collocations and selectional preferences to generate distractors. Evaluations, discussions and related applications of our approaches to the tasks of word sense disambiguation and item generation are presented in Section 6, which will be followed by the concluding section.

2. Data Analysis and Corpus Preparation

2.1 Cloze items for Taiwan College Entrance Examinations

Since our current goal is to generate cloze items for college entrance examinations, we analyzed the effectiveness of considering the linguistic features of cloze items with statistics collected from college entrance examinations administered in Taiwan. We collected and analyzed the properties of the test items used in the 1992-2003 entrance examinations. Among the 220 collected multiple-choice cloze items, the keys to the cloze items that were used in the examinations were only verbs (31.8%), nouns (28.6%), adjectives (23.2%) or adverbs (16.4%). For this reason, we will focus on generating cloze items for these four categories. Moreover, the cloze items contained between 6 and 28 words. Figure 3 depicts the distribution of the number of words in the cloze items. The mean was 15.98, and the standard deviation was 3.84.

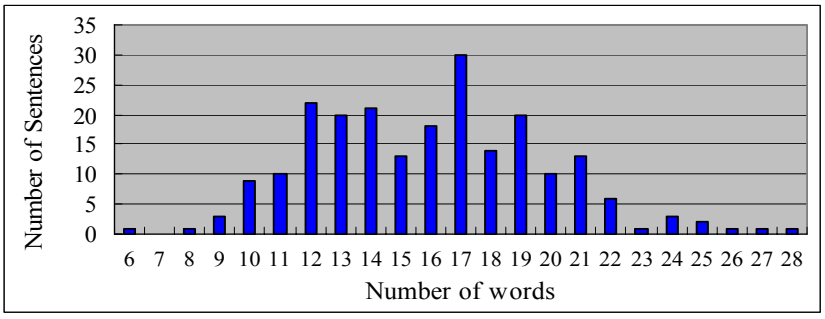

Figure 3. Distribution of the lengths of multiple-choice cloze items

In addition, the Web site of the College Entrance Examination Center provides statistics of testees' responses to a total of 40 multiple-choice cloze items that were used in college entrance examinations held in the years 2002 and 2003 [Taiwan CEEC 2004]. In each of these administrations, more than 110,000 students took the English test. The Web site contains statistics for the error rates of three different groups: ALL, HIGH, and LOW. The ALL group includes all testees, the HIGH group consists of testees whose overall English scores are among the top third, and the LOW group consists of testees whose scores are among the bottom third. Table 1 shows the correlations between the word frequency and selectional-preference (SP, henceforth) strengths of keys and distractors with the error rates observed in different student groups. We will explain how we calculated the frequencies and SP strengths of words in Sections 0 and 4.1, respectively.

From the perspective of correlation, the statistics slightly support our intuition that less frequent words make cloze items more difficult to solve. This claim holds for the ALL and HIGH groups in Table 1. However, the error rates for the LOW group do not correlate with the ranks of word frequency significantly. We suspect that this might be because examinees in the LOW group made more random guesses than average students did. We subtracted the error rates of the HIGH group from the error rates of the LOW group, and computed the correlation between the resulting differences between test items and the ranks of word frequency of the keys in the test items. The results are reported in the DIFF column. The DIFF column shows that using less frequent words in items reduced the items' ability to discriminate between students in the HIGH and LOW groups. The differences in error rates between these groups decreased when less frequent words were used in the cloze items. Figure 4 shows details of the relationships between the error rates and ranks of word frequency of the 40 items that we used to generate Table 1. Since the correlations are not very high, as shown in Table 1, clear trends are not apparent. The charts are included here to allow readers to make their own judgments as to how the error rates and ranks of word frequency are related.

In stark contrast, the correlations shown in the bottom half of Table 1 do not offer a consistent interpretation of the relationship between the error rates of different groups and the SP strengths. The negative numbers in the third row of statistics indicate that, when the SP strengths between the keys and stems increase, the error rates of all groups decrease. This is what one might expect. However, the negative statistics in the last row also suggest that as the SP strengths between the distractors and stems increase, the error rates decrease as well—a phenomenon quite hard to explain. We had expected to see the opposite trend, because distractors should be more misleading when they are more related to the stem. This surprising result might be due to the fact that selectional preference alone is not sufficient to explain students' performance in English tests. Identifying all the factors that can explain students' performance in language tests may require expertise in education, psychology, and linguistics,

which is beyond the expertise of the authors and the scope of this paper. Nevertheless, as we will show shortly, selectional preference can be instrumental in selecting sentences with desired word senses for use in the item-generation task.

Table 1. Correlations between linguistic features and (1) error rates of items for all students (ALL), (2) error rates of items for the top 33% of the students in the English tests (HIGH), (3) error rates of items of the bottom 33% of the students (LOW), and (4) the differences in error rates of items for the LOW and HIGH groups (DIFF)

		ALL	HIGH	LOW	DIFF
rank of word frequency (rank 1 is most frequent)	key	0.07	0.14	-0.07	-0.21
	distractors	0.11	0.15	0.03	-0.15
selectional-preference strength with the stem of the items	key	-0.17	-0.15	-0.07	0.13
	distractors	-0.20	-0.14	-0.21	0.00

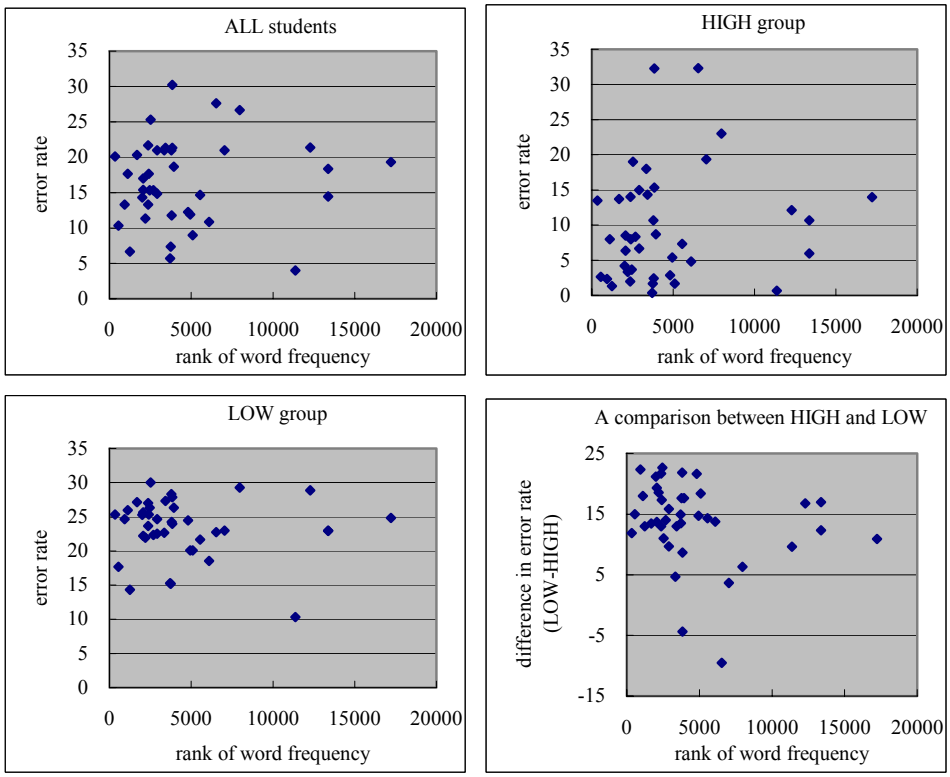

Figure 4. The relationships between error rates and rank of word frequency

2.2 Corpus Preparation and Lexicons

As indicated in Figure 1, a major step in our approach is acquiring sentences from the Web before we produce items. In this pilot study, we retrieved materials from *Studio Classroom* <www.studioclassroom.com>, the *China Post* <www.chinapost.com.tw>, *Taiwan Journal* <taiwanjournal.nat.gov.tw> and *Taiwan Review* <publish.gio.gov.tw> by using a Web crawler. We chose these online journals and news reports partially because they offer up-to-date news at a low spelling error rate and partially because that they can be downloaded at no cost. So far, we have collected in our corpus 163,719 sentences that contain 3,077,474 word tokens and 31,732 word types. Table 2 shows the statistics for verbs, nouns, adjectives, adverbs, and the whole database.

Table 2. Statistics of words in the corpus

	Verbs	Nouns	Adjectives	Adverbs	Overall
Word Tokens	484,673 (16%)	768,870 (25%)	284,331 (9%)	121,512 (4%)	3,077,474 (100%)
Word Types	5,047 (16%)	14,883 (47%)	7,690 (24%)	1,389 (4%)	31,732 (100%)

As a preprocessing step, we look for useful texts from Web pages that are encoded in the HTML format. We need to extract texts from titles, the main bodies of reports, and multimedia contents, and then segment the extracted paragraphs into individual sentences. We segment the extracted texts with the help of Reynar's MXTERMINATOR, which achieved 97.5% precision in segmenting sentences in the Brown and Wall Street Journal corpora [Reynar and Ratnaparkhi 1997]. We then tokenize words in the sentences before assigning useful tags to the tokens. Because we do not employ very precise methods for tokenization, strings may be separated into words incorrectly. Hence, although the statistics reported in Table 2 should be close to actual statistics, the numbers are not very precise.

We augment the texts with an array of tags that facilitate cloze item generation. We assign part-of-speech (POS) tags to words using Ratnaparkhi's MXPOST, which adopts the Penn Treebank tag set [Ratnaparkhi 1996]. Based on the assigned POS tags, we annotate words with their lemmas. For instance, we annotate *classified* with *classify* and *classified*, respectively, when the *classified* has *VBN* (i.e., past participle) and *JJ* (i.e., adjective) as its POS tags. We also mark the occurrences of phrases and idioms in sentences using Lin's MINIPAR [Lin 1998]. This partial parser also allows us to identify such phrases as *arrive at* and *in order to* that appear consecutively in sentences. This is certainly not sufficient for creating items for testing phrases and idioms, and we are currently looking for a better alternative.

MINIPAR mainly provides partial parses of sentences that we can use in our system. With these partial parses, words that are directly related to each other can be identified easily,

and we can apply these relationships between words in word sense disambiguation. For easy reference, we will call words that have a direct syntactic relationship with a word *W* as *W*'s **signal words** or simply **signals**.

After performing these preprocessing steps, we can calculate the word frequencies using the lemmatized texts. As explained in Section 2.1, we consider the most frequent word as the first word in the list, and order the words according to decreasing frequency. Also, as stated in Section 2.1, we focus on creating items for testing verbs, nouns, adjectives, and adverbs, we focus on the signals of words with these POS tags in sentences for disambiguating word senses, and we annotate such information in each sentence.

When we need lexical information about English words, we resort to machine readable lexicons. Specifically, we use WordNet <www.cogsci.princeton.edu/~wn/> when we need definitions and sample sentences of words for disambiguating word senses, and we consult HowNet <www.keenage.com> for information about classes of verbs, nouns, adjectives, and adverbs.

3. System Architecture

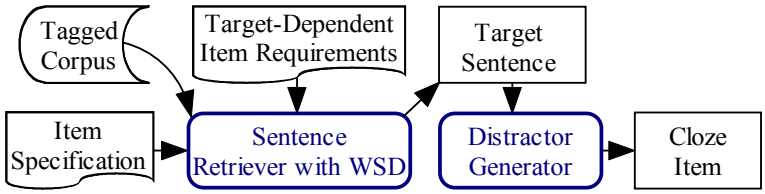

Figure 5. Main components of our cloze-item generator

We create cloze items in two major steps as shown in Figure 5. Constrained by the administrator's *Item Specification* and *Target-Dependent Item Requirements*, the *Sentence Retriever* selects a sentence for a cloze item from a *Tagged Corpus*, which we discussed its preparation in Section 0. Using the *Item Specification*, the test administrator selects the key for the desired cloze item and specifies the part-of-speech and sense of the key that will be used in the item. Figure 6 shows the interface of the *Item Specification*. Our system then attempts to create the requested items. The *Target-Dependent Item Requirements* specify general principles that should be followed in creating items for a particular test. For example, the number of words in cloze items in the college entrance examinations administered in Taiwan ranges from 6 to 28, and one may wish this as a guideline for creating drill tests. In addition, our system allows the test administrator to not specify the key and to request that the system provide a particular number of items for a particular part of speech instead.

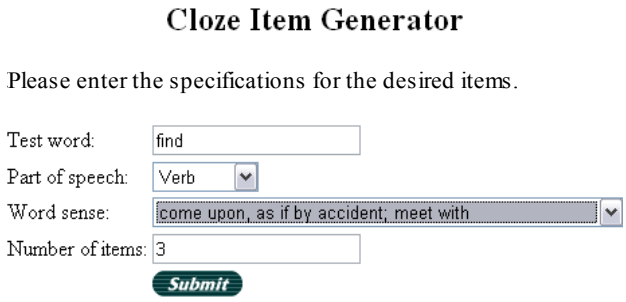

Figure 6. Interface for specifying cloze items

After retrieving the target sentences, the *Distractor Generator* considers such constraining factors as word frequency, collocations, and selectional preferences in selecting distractors. In cases where the generator cannot find sufficient distractors that go with the key and the target sentence, our system abandons the target sentence and starts the process all over again.

4. Target Sentence Retriever

The sentence retriever shown in Figure 5 extracts qualified sentences from the corpus. A sentence must contain the desired key with the requested POS to be considered as a candidate target sentence. We can easily conduct this filtering step using the MXPOS. Having identified a candidate sentence, the item generator needs to determine whether the sense of the key also meets the requirement. We conduct word sense disambiguation based on a generalized notion of selectional preferences.

4.1 Learning Strength of Generalized Selectional Preferences

Selectional preferences refer to the phenomenon that, under normal circumstances, some words constrain the meanings of other words in a sentence. A common illustration of selectional preferences is the case in which the word "chair" in the sentence "Susan interrupted the chair" must denote a person rather than a piece of furniture [Resnik 1997; Manning and Schütze 1999].

We extend this notion to the relationships between a word of interest and its signals, with the help of HowNet. HowNet provides the semantic classes of words; for instance, both *instruct* and *teach* belong to the class of *teach*, and both *want* and *demand* may belong to the class of *need*. Let w be a word of interest, and let π be the word class, defined in HowNet, of a signal of w. We denote the frequency with which both w and π participate in the syntactic relationship, v, as $f_v(w,\pi)$, and we denote the frequency with which w participates in the v relationship in all situations as $f_v(w)$. We define the strength of the selectional preference of

w and π under the relationship v as follows:

$$A_v(w,\pi) = \frac{f_v(w,\pi)}{f_v(w)}. \quad (1)$$

We consider limited types of syntactic relationships. Specifically, the signals of a verb include its subject(noun), object(noun), and the adverbs that modify the verb. Hence, the syntactic relationships for verbs include *verb-object*, *subject-verb*, and *verb-adverb*. The signals of a noun include the adjectives that modify the noun and the verb that uses the noun as its object or predicate. For instance, in "Jimmy builds a grand building," both "build" and "grand" are signals of "building." The signals of adjectives and adverbs include the words that they modify and the words that modify the adjectives and adverbs.

We obtain statistics about the strengths of selectional preferences from the tagged corpus. The definition of $f_v(w)$ is very intuitive and is simply the frequency with which the word w participates in a relationship v with any other words. We initialize $f_v(w)$ to 0 and add 1 to it every time we observe that w participates in a relationship v with any other words.

In comparison, it is more complex to obtain $f_v(w,\pi)$. Assume that s is a signal word that participates in a relationship v with w, and that the POS of s is x in this relationship. When s has only one possible sense under the POS x, and when the main class of this sole sense is π, we increase $f_v(w,\pi)$ by 1. (When HowNet uses multiple fundamental words to describe a sense, the leading word is considered the main class in our computation.) When s itself is polysemous, the learning step is a bit more involved. Assume that s has y possible senses under the POS x, and that the main classes of these senses belong to classes in $\Pi(s) = \{\pi_1,\ldots,\pi_i,\ldots,\pi_y\}$. We increase the co-occurrences of each of these classes and w, $f_v(w,\pi_i)$, i=1,…,y, by 1/y. We distribute the weight for a particular co-occurrence of π_i with w evenly, because we do not have a semantically tagged corpus. With MINIPAR, we only know what syntactic relationship holds between s and w. Without further information or disambiguating the signal words, we choose to weight each sense of s equally. Table 3 shows the statistics, collected from our corpus, for three verbs *eat*, *see* and *find* to take two classes of nouns, *Human* and *Food*, as their objects.

Table 3. Examples of the strengths of selectional preferences, $A_{verb-object}(w,\pi)$

Verb-Object	**Eat**	**See**	**Find**
Human	0.047	0.487	0.108
Food	0.441	0.005	0.057

4.2 Word Sense Disambiguation

We employ generalized selectional preferences to determine the sense of a polysemous word in a sentence. Consider the task of determining the sense of *spend* in the candidate target sentence "*They say film makers don't spend enough time developing a good story.*" The word *spend* has two possible meanings in WordNet.

1. (99) spend, pass – (pass (time) in a specific way; "How are you spending your summer vacation?")
2. (36) spend, expend, drop – (pay out; "I spend all my money in two days.")

Each definition of a possible sense includes (1) the **head words** that summarize the intended meaning, (2) a short explanation, and (3) a sample sentence. When we focus on the disambiguation of a word, we do not consider the word itself as a head word. Hence, *spend* has one head word, i.e., *pass*, in the first sense and two head words, i.e., *extend* and *drop*, in the second sense.

An intuitive method of determining the meaning of *spend* in a target sentence is to replace *spend* in the target sentence with its head words. The head words of the correct sense should fit into the target sentence better than head words of other competing senses. We judge whether a head word fits well into the position of the key based on the SP strength of the head word along with the word class of the signals of the key. Since a sense of the key may include many head words, we define the score of a sense as the average SP strength of the head words of the sense along with all the signal words of the key. This intuition leads to the first part of the total score for a sense, i.e., Ω_t, that we will present shortly.

In addition, we can compare the similarity of the contexts of *spend* in the target sentence and sample sentences, where *context* refers to the classes of the signals of the key being disambiguated. For the current example, we can compare whether the subject and object of *spend* in the target sentence belong to the same classes as the subjects and objects of *spend* in the sample sentences. The sense whose sample sentence offers a more similar context for *spend* in the target sentence receives a higher score. This intuition leads to the second part of the total score for a sense, i.e., Ω_s, that we will present below.

4.2.1 Details of Computing $\Omega_t(\theta_i \mid w,T)$: Replacing Keys with Head Words

Assume that word *w* has *n* senses in the lexicon. Let $\Theta = \{\theta_1,\ldots,\theta_i,\ldots,\theta_n\}$ be the set of senses of *w*. Assume that sense θ_i of word *w* has m_i head words in WordNet. (Note that we do not consider *w* as its own head word.) We use the set $\Lambda_i = \{\lambda_{i,1},\lambda_{i,2},\ldots,\lambda_{i,m_i}\}$ to denote the set of head words that WordNet provides for sense θ_i of word *w*.

When we use the partial parser to parse the target sentence *T* for a key, we obtain information about the signal words of the key. Moreover, when each of these signals is not

polysemous under their current POS tags, we look up their classes in HowNet and adopt the first listed class for each of the signals. Assume that there are $\mu(T)$ signals for the keyword w in a sentence T. We use the set $\Psi(T,w)=\{\psi_{1,T},\psi_{2,T},\ldots,\psi_{\mu(T),T}\}$ to denote the set of signals for w in T. Correspondingly, we use $\nu_{k,T}$ to denote the syntactic relationship between w and $\psi_{k,T}$ in T, and use $\Gamma(T,w)=\{\nu_{1,T},\nu_{2,T},\ldots,\nu_{\mu(T),T}\}$ to denote the set of relationships between signals in $\Psi(T,w)$ and w. Finally, we denote the class of $\psi_{k,T}$ as $\pi_{k,T}$ and the set of classes of the signals in $\Psi(T,w)$ as $\Pi(T,w)=\{\pi_{1,T},\pi_{2,T},\ldots,\pi_{\mu(T),T}\}$.

Recall that Equation (1) defines the strength of the selectional preference between a word and a class of the word's signal. Therefore, the following formula defines the averaged strength of the selectional preference of a head word $\lambda_{i,j}$ of sense θ_i of w with the signal words of w in T:

$$\frac{1}{\mu(T)}\sum_{k=1}^{\mu(T)}A_{\nu_{k,T}}(\lambda_{i,j},\pi_{k,T}).$$

When θ_i contains multiple head words, it is natural for us to compute the average strength of all the head words, excluding w. Hence, (2) measures the possibility of w taking sense θ_i in T. Note that $\Omega_t(\theta_i \mid w,T)$ must fall in the range [0,1] according to the definitions of (1) and (2):

$$\Omega_t(\theta_i \mid w,T)=\frac{1}{m_i}\sum_{j=1}^{m_i}\frac{1}{\mu(T)}\sum_{k=1}^{\mu(T)}A_{\nu_{k,T}}(\lambda_{i,j},\pi_{k,T}). \quad (2)$$

The computation of $\Omega_t(\theta_i \mid w,T)$ in (2) becomes more complicated when a signal, say $\psi_{k,T}$, of the key is polysemous. In this case, we face the problem of disambiguating the contextual information that we rely on for having to disambiguate the key. To terminate this mutual dependence between the senses of the key and the signal words, we use the average SP strength of the signal word in place of $A_{\nu_{k,T}}(\lambda_{i,j},\pi_{k,T})$. Specifically, we assume that $\psi_{k,T}$ has q senses when it participates in the $\nu_{k,T}$ relationship with the key, and we assume that the first listed classes of these q senses of $\psi_{k,T}$ are $\pi_{k,T,1},\pi_{k,T,2},\cdots,\pi_{k,T,q}$. We use the following definition of $A_{\nu_{k,T}}(\lambda_{i,j},\pi_{k,T})$ in Equation (2):

$$A_{\nu_{k,T}}(\lambda_{i,j},\pi_{k,T})=\frac{1}{q}\sum_{r=1}^{q}A_{\nu_{k,T}}(\lambda_{i,j},\pi_{k,T,r}). \quad (3)$$

4.2.2 Details of Computing Ω_s: Comparing the Similarity of Sample Sentences

Since WordNet provides sample sentences for important words, we can use the degrees of similarity between the sample sentences and the target sentence to disambiguate the word sense of the key in the target sentence. Let T and S be the target sentence of w and a sample sentence of sense θ_i of w, respectively. We treat it as a sign of similarity if the signal words that have the same syntactic relationships with the key in both sentences also belong to the

same class. Note specifically that we check the classes of the signal words, rather the signal words themselves. We compute this part of the score, Ω_S, for θ_i using the following three-step procedure. If there are multiple sample sentences for a given sense, say θ_i of *w*, we compute the score in (4) for each sample sentence of θ_i and use the average score as the final score for θ_i.

Procedure for computing $\Omega_S(\theta_i \mid w,T)$

1. We compute the signal words of the key and their relationships with the key in the target and sample sentences as follows:

   $\Psi(T,w) = \{\psi_{1,T}, \psi_{2,T}, \ldots, \psi_{\mu(T),T}\}$ (signal words of *w* in *T*),

   $\Psi(S,w) = \{\psi_{1,S}, \psi_{2,S}, \ldots, \psi_{\mu(S),S}\}$ (signal words of *w* in *S*),

   $\Gamma(T,w) = \{\nu_{1,T}, \nu_{2,T}, \ldots, \nu_{\mu(T),T}\}$ (syntactic relationships of signals words with *w* in *T*),

   $\Gamma(S,w) = \{\nu_{1,S}, \nu_{2,S}, \ldots, \nu_{\mu(S),S}\}$ (syntactic relationships of signals words with *w* in S).

2. We look for a pair $\psi_{j,T}$ and $\psi_{k,S}$ such that $\nu_{j,T} = \nu_{k,S}$, and check whether the matching $\psi_{j,T}$ and $\psi_{k,S}$ belong to the same word class. That is, for each signal of the key in *T*, we check the signals of the key in *S* for matching syntactic relationships (with the key) and word classes, and record the number of matched pairs in $M(\theta_i,T)$ for each θ_i. The matching process is complicated by the fact that signal words can be polysemous as well. When this situation occurs, the credit for each match is recorded proportionally. Assume that the signal word $\psi_{j,T}$ has $n_{j,T}$ possible classes, $\Pi(\psi_{j,T}) = \{\pi_{j,T,1}, \pi_{j,T,2}, \ldots, \pi_{j,T,n_{j,T}}\}$, when it participates in a $\nu_{j,T}$ relationship with *w* in the target sentence *T*. Assume that the signal word $\psi_{k,S}$ has $n_{k,S}$ possible classes, $\Pi(\psi_{k,S}) = \{\pi_{k,S,1}, \pi_{k,S,2}, \ldots, \pi_{k,S,n_{k,S}}\}$, when it participates in a $\nu_{k,S}$ relationship with *w* in a sample sentence *S*. If $\nu_{j,T} = \nu_{k,S}$, then we consider that there is a $1/n_{j,T}$ match whenever a class in $\Pi(\psi_{j,T})$ is matched by a class in $\Pi(\psi_{k,S})$. The pseudo code for computing $M(\theta_i,T)$ is as follows:

   (1) $M(\theta_i,T) = 0$;
   (2) mark all $\nu_{j,T} \in \Gamma(T,w), j = 1, 2, \ldots, \mu(T)$, as unmatched;
   (3) for ($j = 0; j < \mu(T); j++$)
   (4) for ($k = 0; k < \mu(S); k++$)
   (5) if (($\nu_{j,T}$ unmatched) and ($\nu_{j,T} = \nu_{k,S}$))
   (6) for($l = 0; l < n_{j,T}; l++$)
   (7) for($m = 0; m < n_{k,S}; m++$)
   (8) if ($\pi_{j,T,l} = \pi_{k,S,m}$)
   (9) {

(10) mark $v_{j,T}$ as matched;

(11) $M(\theta_i,T) = M(\theta_i,T) + 1/n_{j,T}$

(12) }

3. The following score measures the proportion of matched relationships among all relationships between a sense θ_i of the key and its signals in the target sentence:

$$\Omega_s(\theta_i \mid w,T) = \frac{M(\theta_i,T)}{\mu(T)}. \quad (4)$$

4.2.3 Computing the Final Score for Each Sense

The score for *w* to take sense θ_i in a target sentence *T* is the sum of $\Omega_t(\theta_i \mid w,T)$ defined in (2) and $\Omega_s(\theta_i \mid w,T)$ defined in (4), so the sense of *w* in *T* will be set to the sense defined in (5) when the score exceeds a selected threshold. When the sum of $\Omega_t(\theta_i \mid w,T)$ and $\Omega_s(\theta_i \mid w,T)$ is too small, we avoid making arbitrary decisions about the word senses. There can be many other candidate sentences that include the key, so we can check the usability of these alternatives without having to stick to a sentence that we cannot disambiguate with sufficient confidence. We will discuss and illustrate effects of choosing different thresholds in Section 6.

$$\arg\max_{\theta_i \in \Theta} \quad \Omega_t(\theta_i \mid w,T) + \Omega_s(\theta_i \mid w,T) \quad (5)$$

5. Distractor Generation with Generalized Collocation

Distractors in multiple-choice items influence the possibility of guessing answers correctly. If we use extremely impossible distractors in the items, examinees may be able to identify the correct answers without really knowing the keys. Hence, we need to choose distractors that appear to fit the gap without having multiple possible answers to items in a typical cloze test.

There are principles and alternatives that are easy to implement and follow. Antonyms of the key are choices that average examinees will identify and ignore. The part-of-speech tags of the distractors should be the same as that of the key in the target sentence. Hence, a word will not be a good distractor if it does not have the same part of speech as the key of if it has affixes that indicate its part of speech. We may also take cultural background into consideration. Students with Chinese background tend to associate English words with their Chinese translations. Although this learning strategy works most of the time, students may find it difficult to differentiate between English words that have very similar Chinese translations. Hence, a culture-dependent strategy is to use English words that have similar Chinese translations as the key as distractors.

To generate distractors systematically, we employ word frequency ranks to select words

as candidates [Poel and Weatherly 1997; Wang *et al*. 2003]. Assume that we are generating an item for a key whose part of speech is ρ, that there are n word types whose parts of speech may be ρ in the dictionary, and that the word frequency rank of the key among these n word types is m. We randomly select words whose ranks fall in the range [m-n/10, m+n/10] among these n word types as candidate distractors. These distractors are then screened based on how well they fit into the target sentence, where *fitness* is defined based on the collocations of the word classes of the distractors and other words in the stem of the target sentence.

Since we do not examine the semantics of the target sentences, a relatively safe method for filtering distractors is to choose words that seldom collocate with important words in the target sentence. The "important words" are defined based on the parts of speech of the words and the syntactic structures of the target sentences. Recall that we have marked the words in the corpus with their signal words as discussed in Section 0. Those words that have more signal words in a sentence usually contribute more to the meaning of the sentence, so they should play a more important role in the selection of distractors. In addition, we consider words that have clausal complements in a sentence as important words. Let $T = \{t_1, t_2, \ldots, t_q\}$ denote the set of words, excluding the key, in the target sentence. We therefore define the set of important words $X \subseteq T$ such that each word in X either (1) has two or more signal words in T and is a verb, noun, adjective, or adverb, or (2) has a clausal complement.

Assume that $X \subseteq T$ is the set of important words in T, i.e., $X = \{x_1, x_2, \ldots, x_p\}$, $p \le q$. Let $\Pi(\kappa)$ and $\Pi(x_j)$, respectively, denote the sets of word classes of a candidate distractor κ and an important word x_j. Since we have no semantically tagged corpus, we will judge whether a candidate distractor fits the gap in the test item by checking the co-occurrence of the word class of the distractor and the word classes of the important words in the candidate sentence. A high co-occurrence score will strongly indicate that the candidate distractor is inappropriate.

Let $C = \{S_1, S_2, \ldots, S_{\aleph}\}$ denote the set of sentences in the corpus. We compute the pointwise mutual information between the word classes of a distractor κ and every important word in the target sentence, and take the average as the co-occurrence strength. Let $\zeta(S_i, \kappa)$ denote whether a sentence $S_i \in C$ contains a word whose word classes overlap with the word classes of κ. That is, $\zeta(S_i, \kappa)$ will be either 1 or 0, indicating whether a sense of κ is used in the sentence. Notice that it is not necessary for the word κ itself to be used. We define the probability of occurrence of any word class of κ as follows:

$$\Pr(\Pi(\kappa)) = \frac{1}{\aleph} \sum_{i=1}^{\aleph} \zeta(S_i, \kappa).$$

Analogously, we compute the probability of occurrence of any word class of an important word x_j, $\Pr(\Pi(x_j))$, as follows:

$$\Pr(\Pi(x_j)) = \frac{1}{\aleph}\sum_{i=1}^{\aleph}\zeta(S_i, x_j).$$

In addition, we let $\xi(S_i, \kappa, x_j)$ denote whether a sentence $S_i \in C$ uses a word with a word class in $\Pi(\kappa)$ and another word with a word class in $\Pi(x_j)$. Similar to $\zeta(S_i, \kappa)$, $\xi(S_i, \kappa, x_j)$ is also a Boolean variable. Using this notation, we define the co-occurrence of word classes in $\Pi(\kappa)$ and $\Pi(x_j)$ as follows:

$$\Pr(\Pi(\kappa), \Pi(x_j)) = \frac{1}{\aleph}\sum_{i=1}^{\aleph}\xi(S_i, \kappa, x_j).$$

Having obtained these probability values, we can compute the average pointwise mutual information of a candidate distractor with all of the important words in the target sentence as follows:

$$\mathit{fit}(\kappa) = \frac{-1}{p}\sum_{x_j \in X}\log\frac{\Pr(\Pi(\kappa), \Pi(x_j))}{\Pr(\Pi(\kappa))\Pr(\Pi(x_j))}. \quad (6)$$

We accept candidate words whose scores are better than 0.3 as distractors. The term inside the summation is the pointwise mutual information between κ and x_j, where we consider not the occurrences of the words but the occurrences of their word classes. We negate the averaged sum so that classes that seldom collocate will receive higher scores, thus avoiding multiple answers to the resulting cloze items. We set the threshold to 0.3, based on statistics about (6) that were calculated based on the cloze items administered in the 1992-2003 college entrance examinations in Taiwan.

6. Evaluations, Analyses, and Applications

6.1 Word Sense Disambiguation

Word sense disambiguation is an important topic in natural language processing research [Manning and Schütze 1999]. Different approaches have been evaluated in different setups, and a very wide range of achieved accuracy [40%, 90%] has been reported [Resnik 1997; Wilks and Stevenson 1997]. Hence, objective comparison between different approaches is not a trivial task. It requires a common test environment like SENSEVAL [ACL SIGLEX 2005]. Therefore, we will only present our own results.

Table 4. Accuracy in the WSD task

POS of the key	Baseline	Threshold = 0.4	Threshold = 0.7
Verb	38.0%(19/50)	57.1%(16/28)	68.4%(13/19)
Noun	34.0%(17/50)	63.3%(19/30)	71.4%(15/21)
Adjective	26.7%(8/30)	55.6%(10/18)	60.0%(6/10)
Adverb	36.7%(11/30)	52.4%(11/21)	58.3%(7/12)

We arbitrarily chose 160 sentences that contained polysemous words for disambiguation. A total of 50, 50, 30, and 30 samples were selected for polysemous verbs, nouns, adjectives, and adverbs, respectively. We chose these quantities of sentences based on the relative frequencies of 31.8%, 28.6%, 23.2%, and 16.4% that we discussed in Section 2. We measured the percentages of correctly disambiguated words in these 160 samples, and Table 4 shows the results. In calculating the accuracy, we used the definitions of word senses in WordNet.

The **baseline** column shows the resulting accuracy when we directly used the most common sense, as recorded in WordNet, for the polysemous words. For example, using the definitions of *spend* given in Section 4.2, the first alternative is the default sense of *spend*. The rightmost two columns show the resulting accuracy achieved with our approach when we used different thresholds for applying (5). Our system made fewer decisions when we increased the threshold, as we discussed previously in Section 4.2, and the threshold selection evidently affected the precision of word sense disambiguation evidently. Not surprisingly, a higher threshold led to higher precision, but the rejection rate increased at the same time. For instance, when the threshold was 0.4, our system judged the keys in 28 sentences for verbs, and, when the threshold increased to 0.7, only 19 judgments were made by our system. Out of these 28 and 19 judgments, 16 and 13 were correct, respectively. Sentences whose total scores did not exceed the chosen threshold were simply dropped. Since the corpus can be extended to include more and more sentences, we have the luxury of ignoring sentences and focusing on the precision rather than the rejection rate of the sentence retriever.

6.2 Cloze Item Generation

Figure 7 shows a sample output for the specification shown in Figure 6. Given the generated items, the test administrator can choose the best items via the interface for compiling test questions. Although we have not implemented the post-editing component completely, teachers will be allowed to change the words in the recommended test items and organize the test items according to each teacher's preferences.

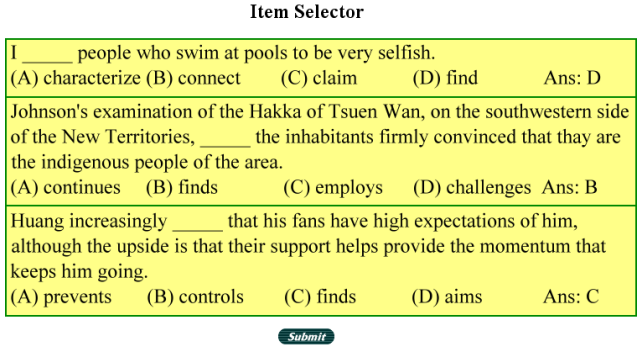

Figure 7. Items generated by the session shown in Figure 6

We asked the item generator to create 200 items in the evaluation. To mimic the distribution of real world examinations, we allocated different numbers of items for verbs, nouns, adjectives, and adverbs based on the proportions of 31.8%, 28.6%, 23.2%, and 16.4% that we reported in Section 2. Hence, we used 77, 62, 35, and 26 items for verbs, nouns, adjectives, and adverbs, respectively, in the evaluation.

Table 5. Correctness of the generated sentences (with the chosen POS tags and senses)

POS of the key	Number of items	% of correct sentences
Verb	77	66.2%
Noun	62	69.4%
Adjective	35	60.0%
Adverb	26	61.5%
Overall		65.5%

In the evaluation, we requested one item at a time and examined whether the sense and part of speech of the key in the generated item really met the requirements. The threshold for using (5) to disambiguate word sense was set to 0.7. The results of this experiment, shown in Table 5, do not differ significantly from those reported in Table 4. For all four major targets of cloze tests, our system was able to return one correct sentence for less than two target sentences it generated. This result is not surprising, as the WSD task is the bottleneck. Putting constraints on the POS would not change the performance significantly. Notice that we generated two different sets of sentences to collect the statistics shown in Tables 4 and 5, so the statistics vary for the same POS.

In addition, we checked the quality of the distractors and marked those items that had only one correct answer as good items. We asked our system to generate another 200 test items and manually determined whether the generated items each had one solution. Table 6 shows that our system was able to create items with unique answers most of the time. It appears that choosing good distractors for adverbs is the most challenging task. Using different adverbs to modify a sentence affects the meaning of the resulting sentence, but it is relatively less likely that using different adverbs as the modifiers will affect the correctness of the sentence. Hence, it is more likely to have multiple possible answers to test items whose keys are adverbs.

Table 6. Uniqueness of answers to the composed test items

Item category	POS of the key	Number of items	Results
Cloze	**Verb**	64	90.6%
	Noun	57	94.7%
	Adjective	46	93.5%
	Adverb	33	84.8%
		overall	91.5%

6.3 Discussion

The head words and sample sentences in the entries of lexicons provide good guidance for word sense disambiguation. Florian and Wicentowski's unsupervised methods that apply information in WordNet and unlabeled corpora are similar to our method, but only the best performer among their methods offers results that are comparable to our results [Florian and Wicentowski 2002]. (We have to note that the comparison made here is based on reported statistics, and that a fair comparison would require using both systems to disambiguate the same set of test data.) Hence, we are quite encouraged by the current performance of our system. Nevertheless, our approach to word sense disambiguation does have the following problems.

We note that not every sense of every word has sample sentences in WordNet. When a sense does not have any sample sentence, this sense will receive no credit, i.e., 0, for $\Omega_s(\theta_i \mid w,T)$. Consequently, our current reliance on sample sentences in the lexicon causes us to discriminate against senses that do not have sample sentences. This is an obvious drawback in our current design, but this problem is not really detrimental or unsolvable. There are usually sample sentences for important and commonly-used senses of polysemous words, so we hope that this discrimination problem will not occur frequently. To solve this problem once and for all, we could customize WordNet by adding sample sentences for all the senses of important words, though we do not imagine that this is a trivial task.

MINIPAR gives only one parse for a sentence, and we have no guarantee of obtaining the correct parses for our sentences. However, this might not be a big problem as our sentences are relatively short. Recall that our system attempts to choose sentences that contained between 6 and 28 words (with an average of about 16 words). Although such short sentences can still be parsed in multiple ways syntactically, short sentences are usually not syntactically ambiguous, and MINIPAR may work satisfactorily.

Using contextual information to disambiguate words is not as easy as we expected. The method reported in this paper is not perfect, and the resulting precision leaves large room for improvement. When we use selectional preference to compute $\Omega_t(\theta_i \mid w,T)$ in (2), we do not attempt to disambiguate the polysemous signal words of the key. We choose to assume that a polysemous signal word will take on each of the possible senses with equal chances in (3). We allow ourselves to avoid the disambiguation of polysemous signal words by this simplifying decision, so introduce errors in the recommended cloze items when the signal words are polysemous. Were the main goal of our research word sense disambiguation, we would have to resort to a more fully-fledged mechanism when a sentence contained multiple ambiguous words. Identifying the topic or the discourse information about the texts from where the target sentences are extracted are possible ways for disambiguating the signal words, and there are quite a few such work in the literature [Manning and Schütze 1999].

An individual sentence that is extracted from a larger context, e.g., a paragraph, may not contain sufficient information for students to understand the extracted sentence. If understanding the target sentence requires information not contained in the target sentence, it will not be a good idea to use this sentence as a test item, because this extra factor may introduce unnecessary noise that prevents students from answering the test item correctly.

Dr. Lee-Feng Chien of Academia Sinica has pointed out that our use of the sense definitions in WordNet may have demanded unnecessary quality for the word sense disambiguation task. WordNet includes more fine-grained differentiations of senses that may exceed the needs of ordinary learners of English.

The aforementioned weaknesses should not overshadow the viability of our approach. The experimental results obtained in our pilot study indicate that, with our method, one can implement a satisfactory cloze item generator at relatively low cost. Although we must admit that the weaknesses of our approach could become problems if we targeted at a fully automatic item generation [Bejar *et al.* 2003], we suspect that a fully automatic item generator would offer items of appropriate quality for our current application. In our approach to generating cloze items, a reasonable error rate in the word sense disambiguation task is tolerable because human experts will review and select the generated items anyway. As long as we can confine the error rates within a limited range, the computer-assisted generation process will increase the overall productivity.

6.4 More Applications

We have used the generated items in real tests in a freshman-level English class at National Chengchi University, and have integrated the reported item generator in a Web-based system for learning English [Gao and Liu 2003]. In this system, we have two major subsystems: an authoring subsystem and an assessment subsystem. Using the authoring subsystem, test administrators can select items through the interface shown in Figure 7, save the selected items to an item pool, edit the items, including their stems if necessary, and finalize the selection of items for a particular examination. Using the assessment subsystem, students can answer the test items via the Internet, and receive scores immediately if the administrators choose to provide them. Student's answers are recorded for student modeling and for the analysis of item facility and item discrimination.

In addition to supporting cloze tests, our system also can create items for testing idioms and phrases. Figure 8 shows the output of this function. However, we can only support consecutive phrases at this moment. Moreover, we are currently expanding our system to help students with listening and dictation in English [Huang *et al.* 2005]. Our long-term plans are to expand our system to support more aspects of learning English and to enable our system to adapt to students' competency [Liu 2005].

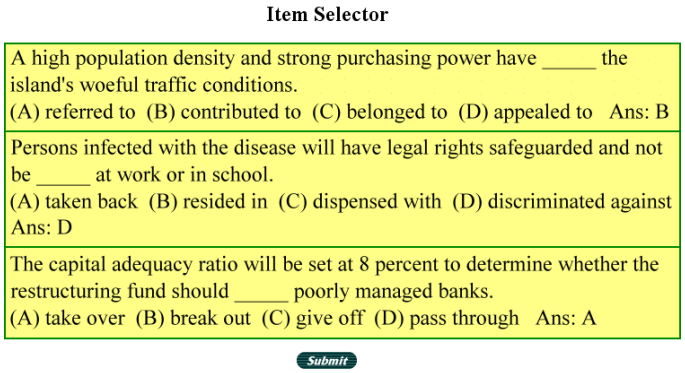

Figure 8. Sample items for testing English phrases

7. Concluding Remarks

Natural language processing techniques prove to be instrumental for creating multiple-choice cloze items that meet very specific needs of test administrators. By introducing word sense disambiguation into the item generation process, we enable each generated cloze item to include words that carry the desired senses. Word sense disambiguation itself is not a trivial task and has been studied actively for years. Although our approach does not lead to perfect selections of the word senses in target sentences, its performance is comparable to that of some modern methods for word sense disambiguation, and we have shown that it can provide crucial aid in the item generation task. After all, it is well known that word sense disambiguation may require information about contexts that cover more than just individual sentences, and that high-quality disambiguation within an individual sentence can be very difficult, if not impossible to achieve [Manning and Schütze 1999].

We have also proposed a new approach to selecting distractors for multiple-choice cloze items. Using the proposed collocation-based measure and word frequencies, we are able to identify distractors that are similar in challenge level with the key of the item, while guaranteeing that there is only one answer to the item about 90% of the time.

Since test administrators can request our system to return multiple items and manually select the best ones for composing tests, it is not absolutely necessary for us to create a perfect item generator. Our system currently generates one usable cloze item for every 1.6 generated items. Nevertheless, we intend to improve this result by considering more advanced linguistic features in sense disambiguation, and will update the results in the near future.

Acknowledgements

We thank Dr. Lee-Feng Chien, the editors of this special issue, and the anonymous reviewers for their invaluable comments on previous versions of this paper. We must admit that we have

not been able to follow all their suggestions for improving this short article, and we are responsible for any remaining problems in this paper. The authors would also like to thank Professor I-Ping Wan of the Graduate Institute of Linguistics of National Chengchi University for adopting test items generated by our system in her English classes in the 2003 Fall semester. This research was supported in part by grants 91-2411-H-002-080, 92-2213-E-004-004, 92-2411-H-002-061, 93-2213-E-004-004, and 93-2411-H-002-013 from the National Science Council of Taiwan. This paper is an expanded version of [Wang *et al.* 2004a] and [Wang *et al.* 2004b].

References

ACL SIGLEX, SESEVAL homepage, http://www.senseval.org/, 2005.

Bejar, I. I., R. R. Lawless, M. E. Wagner, R. E. Bennett and J. Revuelta, "A Feasibility Study of On-the-Fly Item Generation in Adaptive Testing," *Journal of Technology, Learning and Assessment*, 2, 2003, http://www.jtla.org/.

Computational Linguistics, Special issue on word sense disambiguation, 24(1), 1998.

Coniam, D., "A Preliminary Inquiry into Using Corpus Word Frequency Data in the Automatic Generation of English Language Cloze Tests," *Computer Assisted Language Instruction Consortium*, 16(2–4), 1997, pp. 15–33.

Deane, P. and K. Sheehan, "Automatic Item Generation via Frame Semantics," Educational Testing Service: http://www.ets.org/research/dload/ncme03-deane.pdf, 2003.

Dennis, I., S. Handley, P. Bradon, J. Evans and S. Nestead, "Approaches to Modeling Item Generative Tests," *Item Generation for Test Development*, ed. by Irvine and Kyllonen, 2002, pp. 53–72.

Edmonds, P., R. Mihalcea, and P. Saint-Dizier, editors, *Proceedings of the ACL 2002 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Association for Computational Linguistics, 2002.

Gao, Z.-M. and C.-L. Liu, "A Web-Based Assessment and Profiling System for College English," In *Proceedings of the Eleventh International Conference on Computer Assisted Instruction*, 2003, CD–ROM.

Huang, S.-M., C.-L. Liu and Z.-M. Gao, "Computer-Assisted Item Generation for Listening Cloze and Dictation Practice in English," In *Proceedings of the Fourth International Conference on Web-Based Learning*, 2005, forthcoming.

Irvine, S. H. and P. C. Kyllonen, editors, *Item Generation for Test Development*, Lawrence Erlbaum Associates, 2002.

Kornai, A. and B. Sundheim, editors, *Proceedings of HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing*, Association for Computational Linguistics, 2003.

Liu, C.-L., "Using Mutual Information for Adaptive Item Comparison and Student Assessment," *Journal of Educational Technology & Society*, 8(4), 2005, forthcoming.

Lin, D., "Dependency-Based Evaluation of MINIPAR," In *Proceedings of the Workshop on the Evaluation of Parsing Systems in the First International Conference on Language Resources and Evaluation*, 1998.

Manning, C. D. and H. Schütze, *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, 1999.

Oranje, A., "Automatic Item Generation Applied to the National Assessment of Educational Progress: Exploring a Multilevel Structural Equation Model for Categorized Variables," Educational Testing Service: http://www.ets.org/research/dload/ncme03-andreas.pdf, 2003.

Poel, C. J. and S. D. Weatherly, "A Cloze Look at Placement Testing," *Shiken: JALT* (Japanese Association for Language Teaching) *Testing & Evaluation SIG Newsletter*, 1(1), 1997, pp. 4–10.

Ratnaparkhi, A., "A Maximum Entropy Part-of-Speech Tagger," In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1996, pp. 133–142.

Resnik, P., "Selectional Preference and Sense Disambiguation," In *Proceedings of the Applied Natural Language Processing Workshop on Tagging Text with Lexical Semantics: Why, What and How*, 1997, pp. 52–57.

Reynar, J. C. and A. Ratnaparkhi, "A Maximum Entropy Approach to Identifying Sentence Boundaries," In *Proceedings of the Conference on Applied Natural Language Processing*, 1997, pp. 16–19.

Sheehan, K. M., P. Deane, and I. Kostin, "A Partially Automated System for Generating Passage-Based Multiple-Choice Verbal Reasoning Items," paper presented at the National Council on Measurement in Education Annual Meeting, 2003.

Stevens, V., "Classroom Concordancing: Vocabulary Materials Derived from Relevant Authentic Text," *English for Specific Purposes*, 10(1), 1991, pp. 35–46.

Taiwan College Entrance Examination Center (CEEC): Statistics about 2002 and 2003 entrance examinations, 2004, http://www.ceec.edu.tw/exam/e_index.htm/.

van der Linden, W. J. and R. K. Hambleton, editors, *Handbook of Modern Item Response Theory*, Springer, New York, USA, 1997.

van der Linden, W. J. and C. A. W. Glas, editors, *Computerized Adaptive Testing: Theory and Practice*, Kluwer, Dordrcht, Netherlands, 2000.

Wang, C.-H., C.-L. Liu and Z.-M. Gao, "Toward Computer Assisted Item Generation for English Vocabulary Tests," In *Proceedings of the 2003 Joint Conference on Artificial Intelligence, Fuzzy Systems, and Grey Systems*, 2003, CD–ROM.

Wang, C.-H., C.-L. Liu and Z.-M. Gao, "利用自然語言處理技術自動產生英文克漏詞試題之研究," In *Proceedings of the Sixteenth Conference on Computational Linguistics and Speech Processing*, 2004a, pp. 111–120. (in Chinese)

Wang, C.-H., C.-L. Liu and Z.-M. Gao, "Using Lexical Constraints for Corpus-Based Generation of Multiple-Choice Cloze Items," In *Proceedings of the Seventh IASTED*

International Conference on Computers and Advanced Technology in Education, 2004b, pp. 351–356.

Wilks, Y. and M. Stevenson, "Combining Independent Knowledge Sources for Word Sense Disambiguation," In *Proceedings of the Conference on Recent Advances in Natural Language Processing*, 1997, pp. 1–7.

Computational Linguistics and Chinese Language Processing
Vol. 10, No. 3, September 2005, pp. 329-346

Collocational Translation Memory Extraction Based on Statistical and Linguistic Information

Thomas C. Chuang*, Jia-Yan Jian+, Yu-Chia Chang+ and Jason S. Chang+

Abstract

In this paper, we propose a new method for extracting bilingual collocations from a parallel corpus to provide phrasal translation memories. The method integrates statistical and linguistic information to achieve effective extraction of bilingual collocations. The linguistic information includes parts of speech, chunks, and clauses. The method involves first obtaining an extended list of English collocations from a very large monolingual corpus, then identifying the collocations in a parallel corpus, and finally extracting translation equivalents of the collocations based on word alignment information. Experimental results indicate that phrasal translation memories can be effectively used for computer assisted language learning (CALL) and computer assisted translation (CAT).

Keywords: Bilingual Collocation Extraction, Collocational Translation Memory, Collocational Concordancer

1. Introduction

Example-based machine translation (EBMT) has been proposed as an alternative approach to automatic translation. Translations of examples range from two-word to multi-word, translations, with or without syntactic or semantic structures [Nagao 1984; Kitano 1993; Smadja 1993; Lin 1998; Andrimanankasian *et al*. 1999; Carl 1999; Brown 2000; Pearce 2001; Seretan *et al.* 2003]. In the approach, text and translations are preprocessed and stored in a translation memory, which serves as an archive of existing translations that the MT system can reuse. A number of proposed applications for machine translation and computer assisted translation systems use translation examples found in bilingual corpora; these methods include

* Department of Computer Science and Information Engineering, Vanung University, No. 1, Vanung Road, Jhongli, Taoyuan, Taiwan
E-mail: tomchuang@msa.vnu.edu.tw
+ Department of Computer Science, National Tsing Hua University, 101, Kuangfu Road, Hsinchu, Taiwan

[Received January 31, 2005; Revised August 24, 2005; Accepted August 26, 2005]

[Transit 2005], [Deja–Vu 2005], [TransSearch 2005], and [TOTALrecall 2005].

Statistical methods have been proposed for automatic acquisition of bilingual collocations [Smadja *et al.* 1996; Gao *et al*. 2002; Wu and Zhou 2003] from parallel bilingual corpora [Kupiec 1993; Smadja *et al.* 1996; Echizen-ya *et al*. 2003] or from two comparable monolingual corpora [Lu and Zhou 2004]. These bilingual collocations, if acquired in quantity, can enable a machine translation system to produce more native-speaker like translations. However, parallel corpora of substantial size are harder than monolingual corpora to come by. Therefore, in small- to mid-size parallel texts, collocations may not have high enough counts for a statistical method to reliably extract them.

Consider the example of extracting verb-noun collocations for the noun "influence" from 50,000 bilingual sentences (SMEC-50000) in the Sinorama Mandarin-English Corpus (SMEC)[1]. Some useful bilingual collocations in SMEC have very low occurrence counts. For instance, the bilingual collocation "use influence; 發揮 影響力" appears only once in SM-50000 (see Example 1). Such collocations may not be extracted by the methods proposed in the literature.

(1) These circumstances make it unlikely that APEC will be able to avoid reform. In Lai's analysis, the implosion of the Asian economies last year demonstrates their interconnectedness. Therefore, in order to place its own existence on a more secure foundation, Taiwan should carefully observe changes in APEC and **use** its **influence** to make the organization into a vehicle driving regional consolidation.
他分析，亞洲國家近年來經濟危機持續不去，證明瞭彼此的連動性，因此台灣應該注意觀察APEC的轉變，**發揮**意見的**影響力**，以使APEC能夠成爲區域整合的火車頭，爲我國創造更大的生存利基。

A good way of extracting such bilingual collocation might be to first extract "use influence" as a collocation in a large, separate, monolingual corpus, and then identify its instances and translations in the given parallel corpus (e.g., the Sinorama Mandarin-English Corpus). At present, it is not difficult to obtain a much larger monolingual corpus (e.g., the British National Corpus) that contains enough instances of "use influence" such that extraction of such a collocation type is mostly effective. Example (2) shows one of the 60 instances of

[1] The Sinorama Mandarin-English Corpus was originally a database of some 6,000 bilingual articles appearing in Sinorama Magazine dated 1976 to 2001 (Copyright ©2001 Sinorama Magazine & Wordpedia.com Co.) The database is a parallel text corpus, now available from The Association for Computational Linguistics and Chinese Language Processing.

"use influence" in BNC. Such a relatively high count makes it very easy to identify "use influence" as a collocation by using any the method proposed in the literature.

(2) I don't know who it and apparently he asked him that, are, are any of your men gonna be there and if there are he said, I'm, I'm gonna pull out and **use** all my **influence** to stop the march and the IRA police said no there would not be any gunmen there so I thought yeah, fucking right, oh yeah that's easy to say, and then if like the reporter said and, and you believe him and you have the feeble excuse towards a small community he said, you know what 's going on.

We will present a new method that automatically performs shallow parsing on an English corpus to identify all the statistically significant collocation types and their instances in the monolingual corpus and the English part of the given bilingual corpus. After that, the translation of each collocate of each collocation is identified based using primarily the word alignment technique. We will also present a computer assisted translation system, *Tango,* which accepts user queries of words, parts of speech, and types of collocation, and displays citations with bilingual collocations highlighted. An example of a Tango search for bilingual collocations for the noun "influence" is shown in Figure 1.

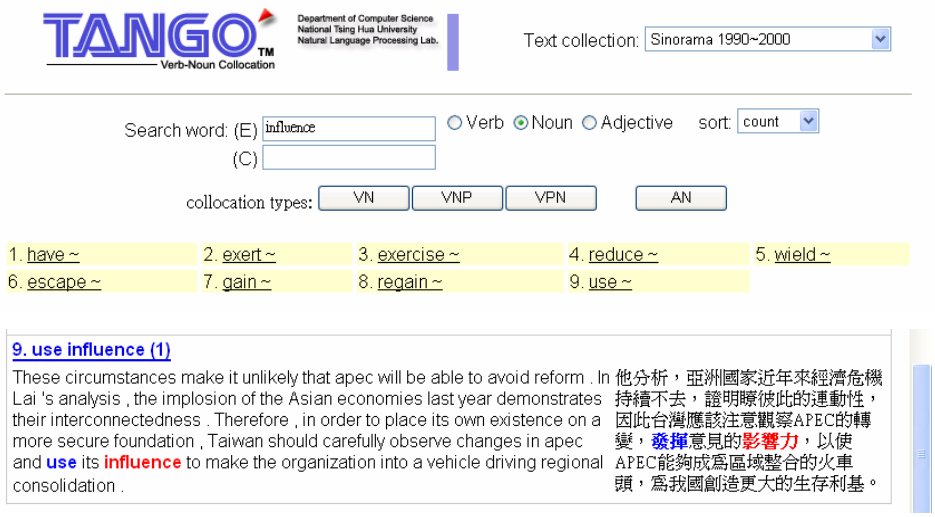

Figure 1. An example of a Tango search for bilingual collocations for the noun "influence".

The rest of the article is organized as follows. We will review related works in the next section. Then we will present our method for automatically processing sentences in monolingual and parallel corpora, and extracting bilingual collocations (Section 3). As part of our evaluation, we will describe an implementation of the proposed approach using SMEC and BNC (Section 4) and discuss the results of our evaluation carried out to assess the performance of bilingual collocation extraction (Section 5).

2. Extraction of Collocational Translation Memory

It is difficult to extract bilingual collocations from parallel corpora due to their limited availability and small sizes. Methods proposed in previous works typically extract collocations based solely on co-occurrence counts and statistical association measures in bilingual corpora. Unfortunately, a substantial part of the collocations in a modest-sized parallel corpus might not have high enough frequency counts for statistical extraction methods to be effective. Many bilingual collocations useful for machine translation and computer assisted language learning may appear only once or twice in a small to medium size corpus. To extract bilingual collocations, a promising approach is to acquire collocations in from a very large monolingual corpus and obtain translations from a parallel corpus.

2.1 Problem Statement

We will focus here on the first step of building a translation memory for a bilingual collocation tool this involves; extracting a set of bilingual collocations instances from a sentence-aligned parallel corpus. These collocation instances can be used for the purpose of computer assisted translation and language learning. Thus, the collocations, including those that appear only once or twice, should be identified in the part of the parallel corpus that is written in one of the two languages. At the same time, it is crucial that the translations also be identified. Therefore, our goal is to return a reasonable-size set of documents that, at the same time, must contain an answer to the question. For simplicity, we will focus on verb-noun collocations in this paper. We formally state the problem that we are addressing below.

Problem Statement: Given a parallel corpus *PC* of *n* pairs of sentences (SE_i, SF_i) written in the first language *E* and the second language *F*. Our goal is to identify a set of *k collocations* and *translations* (CE_{ij}, CF_{ij}) in (SE_i, SF_i). To accomplish this task, we make use of a large corpus *M* with *m* sentences ME_i of texts written in *E* to help identify CE_{ij} in SE_i.

We attempt to identity collocations in a parallel corpus by leveraging another larger monolingual corpus in which collocations appears with higher occurrence counts. Our method is shown in Figure 2.

(1) Annotate the English Sentences with parts of speech, chunk, and clause information (Section 2.2)
(2) Extract English collocation types in ME_i (Section 2.3)
(3) Extract English collocation instances in CE_{ij} in SE_i (Section 2.3)
(4) Identify translation equivalents CF_{ij} to collocation CE_{ij} in SF_i (Section 2.4)

Figure 2. Outline of the process used to extract bilingual collocations

2.2 Taggers for parts of speech, chunks, and clauses

Using an annotated corpus with texts written in *E,* we can develop a tagger based on three Hidden Markov Models: one each for parts of speech, chunks (i.e. basis phrases), and clauses. The training corpus consists of sentences with three levels of annotation for each word: parts of speech, chunk, and clause. Figure 3 shows three levels of annotation for Example (3):

w_i	t_i	u_i	v_i
This	DT	B–NP	S
is	VBZ	B–VP	X
the	DT	B–NP	X
11th	JJ	I–NP	X
consecutive	JJ	I–NP	X
quarter	NN	I–NP	X
in	IN	B–PP	S
which	WDT	B–NP	X
the	DT	B–NP	S
company	NN	I–NP	X
has	VBZ	B–VP	X
paid	VBN	I–VP	X
shareholders	NNS	B–NP	X
an	DT	B–NP	X
extra	JJ	I–NP	X
dividend	NN	I–NP	X
of	IN	B–PP	X
five	CD	B–NP	X
cents	NNS	I–NP	X
.	.	O	X

Figure 3. Examples of three levels of tagging performed on the sentence "This is the 11th consecutive quarter in which the company has paid shareholders an extra dividend of five cents."

(3) This is the 11th consecutive quarter in which the company has paid shareholders an extra dividend of five cents.

As shown in Figure 3, each word is tagged with a parts of speech tag (e.g., DT for determiner and VBZ for third person singular verb), a chunk tag indicating the basis phrase type (e.g., noun phrase, NP, verb phrase VP, etc.) Plus the position of the word (e.g., "B" for words beginning a chunk and "I" for all other words in the chunk), and a clause tag (e.g., "S" for a clause-beginning word and "X" otherwise). The chunk annotation shown in Figure 3 indicates that "This," "the 11th consecutive quarter," "the company," "shareholder," "an extra dividend," and "five cents" are nouns on noun phrases, while "is" and "has paid" are verb phrases. Similarly, the clause annotation results indicate that "This is the 11th consecutive quarter" and "in which the company has paid shareholders an extra dividend of five cents" are the only two clauses in the sentence. With a substantial amount of annotated sentences like the above we can develop three taggers for each level of analysis.

For the parts of speech tagger, the HMM operates on a set of states represented by all possible POS tage, goes through a sequence of state t_i, and produces words w_i, i = 1 to n. An example of transition probability function $P(u_i \mid u_{i-1})$ for the chunk tagger is shown in Figure 4. A first order HMM is characterized by the emission probability, $P(w_i \mid t_i)$, and state transition probability, $P(t_i \mid t_{i-1})$. An example of the emission probability function $P(t_i \mid u_i)$ for the chunk tagger is shown in Figure 5.

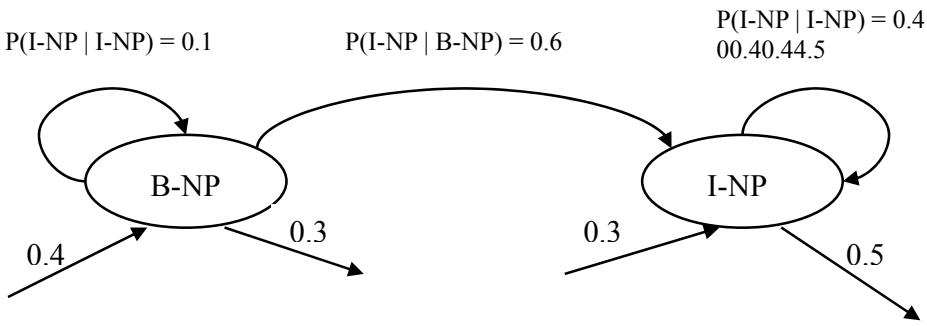

Figure 4. Example of transition probability function $P(u_i \mid u_{i-1})$ ***for the chunk tagger***

POS / state	DT	NN	Others
B-NP	0.6	0.2	0.2
I-NP	0.1	0.7	0.2
others	…	…	…

Figure 5. Example of the emission probability function $P(t_i \mid u_i)$ ***for the chunk tagger***

Specifically, we have

$$P(w) = P(w_1 w_{2 \ldots} w_n) = P(t_1) P(w_1 \mid t_1) \prod_{i=2,n} P(t_i \mid t_{i-1}) P(w_i \mid t_i) \quad . \tag{1}$$

Therefore, we can to derive the parts of speech $t = (t_1 t_{2 \ldots} t_n)$ for the sentence $w = (w_1 w_{2 \ldots} w_n)$ by calculating $(t_1 t_{2 \ldots} t_n)$ that maximizes $P(t \mid w)$. Thus, we have

$$\begin{aligned}(t_1 t_{2 \ldots} t_n) &= \text{argmax}_t P(t \mid w) = \text{argmax}_t P(w \mid t) P(t)/P(w) = \text{argmax}_t P(w \mid t) P(t) \\ &= \text{argmax}_t P(t_1) P(w_1 \mid t_1) \prod_{i=2,n} P(t_i \mid t_{i-1}) P(w_i \mid t_i) \quad .\end{aligned} \tag{2}$$

Similarly, we can derive the chunk tags $(u_1 u_{2 \ldots} u_n)$ and clause tags $(v_1 v_{2 \ldots} v_n)$ for the given sentence $(w_1 w_{2 \ldots} w_n)$ using Equations (3) and (4):

$$(u_1 u_{2 \ldots} u_n) = \text{argmax}_u P(u_1) P(t_1 \mid u_1) \prod_{i=2,n} P(u_i \mid u_{i-1}) P(t_i \mid u_i) \quad , \tag{3}$$

$$(v_1 v_{2 \ldots} v_n) = \text{argmax}_v P(v_1) P(u_1 \mid v_1) \prod_{i=2,n} P(v_i \mid v_{i-1}) P(u_i \mid v_i) \quad . \tag{4}$$

The optimal values of parts of speech tags $(t_1 t_{2 \ldots} t_n)$, chunk tags $(u_1 u_{2 \ldots} u_n)$, and clause tags $(v_1 v_{2 \ldots} v_n)$ can be derived by using a dynamic procedure [Manning and Shutze 1999]. The tagging process is carried out for the *n* source sentences SE_i in a given parallel corpus *PC* and in the *m* sentences ME_i in a large corpus *M*. We will describe in Section 3 how the training data provided the results from the common task CoNLL-2000 and CoNLL-2001 shared tasks (CoNLL, 2000) can be used to estimate the probabilistic functions involved, including $P(t_1)$, $P(t_i \mid t_{i-1})$, $P(w_i \mid t_i)$, $P(u_1)$, $P(u_i \mid u_{i-1})$, $P(t_i \mid u_i)$, $P(v_i)$, $P(v_i \mid v_{i-1})$ and $P(u_i \mid v_i)$.

2.3 Extraction of Collocation Types in *M*

With the chunk and clause tags for the sentences in the monolingual corpus *M*, we can proceed to extract a set of verb-noun collocation types from *M*. To that end, we can consider the heads of phrases in three prevalent verb-noun collocation structures in the corpus: VP+NP, VP+PP+NP, and VP+NP+PP. To extract verbs and nouns that appear in a predicate-object relation, we need to have a full parse of the sentences. However, a state-of-the-art parser can not produce a full parse of unrestricted texts with a very high precision rate. Therefore, we simply assume that a noun phrase following a verb phrase is in a predicate-object relationship unless they belong to two different clauses.

For instance, consider the sentence shown in Example (4):

(4) Confidence in the pound is widely expected to take another sharp dive if trade figures for September due for release tomorrow fail to show a substantial improvement from July and August's near-record deficits.

The taggers described in Section 2.2 will produce the parts of speech tags, chunk tags, and clause tags shown in Examples (5)-(7):

(5) Confidence/NN in/IN the/DT pound/NN is/VBZ widely/RB expected/VBN to/TO take/VB another/DT sharp/JJ dive/NN if/IN trade/NN figures/NNS for/IN September/NNP ,/, due/JJ for/IN release/NN tomorrow/NN ,/, fail/VB to/TO show/VB a/DT substantial/JJ improvement/NN from/IN July/NNP and/CC August/NNP 's/POS near-record/JJ deficits/NNS ./.

(6) Confidence /B-NP in/B-PP the/B-NP pound/I-NP is/B-VP widely/I-VP expected/I-VP
to/I-VP take/I-VP another/B-NP sharp/I-NP dive/I-NP if/B-SBAR trade/B-NP figures/I-NP
for/B-PP September/B-NP due/ADJP for/B-PP release/B-NP tomorrow/I-NP ,/O
fail/O to/O show/O a/B-NP substantial substantial/I-NP improvement/I-NP from/B-PP
July/B-NP and/O August/B-NP 's/B-NP near-record/I-NP deficits/I-NP ./O

(7) Confidence /S in/X the/X pound/X is/X widely/X expected/X
to/S take/X another/X sharp/X dive/X if/S trade/X figures/X
for/X September/X due/X for/X release/X tomorrow/X ,/ *
fail/S to/X show/X a/X substantial/X improvement/X from/X
July/X and/X August/X 's/X near-record/X deficits/X ./X

The words in the same chunk can be further grouped together (as shown in Figure 6) to make it easy to examine the phrase types of two adjacent chunks and extract the head word of each phrase. For instance, we can extract a VN pair (e.g., "*take*" and "*dive*") from an annotated sentence by taking the last words of two adjacent VP and NP chunks.

Phrase	Type
Confidence	NP
In	PP
the pound	NP
is widely expected to ***take***	VP
another sharp ***dive***	NP
If	SBAR
trade figures	NP
For	PP
September	NP

Figure 6. Example of grouping words in a chunk together record by record

Care should be taken to avoid extracting verbs and nouns from two clauses in a sentence. For instance, in Example (8), in some cases, considering only chunk information is not sufficient. For example, the VN pair ("think," "people") taken from the two separate clauses "why do you think" and "people cannot see the top of the building on some days" should be excluded from consideration.

(8)Why do you think people cannot see the top of the building on some days?

Some VN pairs extracted at this stage are free combinations, while some recur more frequently than is likely according to chance and should be considered collocations. After obtaining a list of instances of candidate collocations, we proceed to find distinct collocation types and calculate their word counts and phrase counts in order to verify whether each of them is a valid collocation. After that, we calculate the strength of association between each verb-noun pair in the collocations by using Logarithmic Likelihood Ratio (LLR) statistics. Equation (5) is the formula that computes the LLR.

$$LLR(x;y) = -2\log_2 \frac{{p_1}^{k_1}(1-p_1)^{n_1-k_1}\,(1-p_2)^{n_2-k_2}}{p^{k_1}(1-p)^{n_1-k_1}\;p^{k_2}(1-p)^{n_2-k_2}}; \tag{5}$$

k_1: count of sentences that contain x and y simultaneously;

k_2: count of sentences that contain x but do not contain y;

n_1: count of sentences that contain y;

n_2: count of sentences that do not contain y;

$p_1 = k_1 / n_1;$

$p_2 = k_2 / n_2;$

$p = (k_1+k_2) / (n_1+n_2).$

We have described a method for handling VN collocations. This method can be easily extended to handle VPN and VNP collocations as well. The idea is quite simple. After identifying a VN collocation type where the verb and noun are separated by a preposition, we go on to consider the preposition that comes between the verb and the noun or that follows the verb and noun. The VPN and VNP collocations are validated again by calculating the LLR between each VN pair and the preposition.

2.4 Extraction of Collocation Instances in *PC*

We subsequently identify collocation instances in the n sentence SE_i of the give parallel corpus *PC*. First, each sentence SE_i is subjected to the same POS, chunk, and clause analyses as is applied to the corpus *M*. The collocation instances of the forms VN, VPN, and VNP are extracted in a similar way to that described in Section 2.2. There are two cases in which a collocation instance will be considered as a valid collocation:

1. if it passes the LLR threshold calculated based on the counts of words and co-occurrences in *PC*;
2. if it is in the list of valid collocations found in *M*.

The quantity and quality of collocations in a very large monolingual corpus surely will facilitate collocation identification in a smaller bilingual corpus with better statistical measures.

2.5 Extracting Collocation Translation Equivalents in a Bilingual Corpus

After instances that are most likely valid collocations are obtained from a bilingual corpus, we go on to work on the second part of the parallel corpus *PC*. We exploit statistical word-alignment techniques [Melamed 1997] and dictionaries to find translation candidates for each of the words in a given collocation. Using Melamed's approach, we can establish a word translation based on corpora to supplement English-Chinese dictionaries, which generally suffer due to insufficient information. We first locate the translation of the noun. Subsequently, we locate the verb nearest to the noun translation to find the translation of the verb. Figure 7 shows some examples.

English sentence	Chinese sentence
If in this time no one shows concern for them, and directs them to correct thinking, and teaches them how to express and ***release emotions***, this could very easily leave them with a terrible personality complex	如果這時沒有人關心他們，引導他們正確思考，教他們表達、宣洩**情緒**，極易在人格成長上留下一個打不開的死結。

they can never resolve.	
Occasionally some kungfu movies may appeal to foreign **audiences**, but these too are exceptions to the rule.	偶爾有一些武打片對某些外國**觀眾**有吸引力，但也是個案。

Figure 7. Examples of identifying translations of nouns (in bold) and verbs (shaded) of VN collocation instances in bilingual sentence pairs

3. Implementation and evaluation

We have implemented a program for extracting bilingual collocations based on the proposed method and experimented with 50,000 bilingual sentences (SMEC-50000) from the Sinorama Mandarin-English Corpus (SMEC). We wanted to assess the performance of the program and verify whether useful bilingual collocations in SMEC with very low occurrence counts (e.g., "use influence; 發揮 影響力") could be extracted. Such collocations are beyond the reach of methods previously proposed in the literature.

We used the Brown corpus to develop a parts-of-speech tagger and the CoNLL-2000 benchmark database to build a chunk tagger and clause tagger. The chunk tagger relied on the transition and output probabilities of chunks. Figures 8 and 9 show examples of these two processes. The average precision rate of the chunk tagger was about 93.7%, based on CoNLL testing data.

Chunk tag u_i	Chunk tag u_{i+1}	adj. count($u_i u_{i+1}$)	count(u_i)	P($u_i \mid u_{i+1}$)
B-NP	I-NP	46327.3	67503	0.686300
B-NP	B-VP	8762.3	67503	0.129806
B-NP	O	5418.3	67503	0.080268
B-NP	B-PP	3878.3	67503	0.057454
B-NP	B-NP	1974.3	67503	0.029248
B-NP	B-ADVP	645.3	67503	0.009560
B-VP	I-VP	9830.3	26125	0.345313
B-VP	B-NP	9021.3	26125	0.097619
B-VP	B-PP	2550.3	26125	0.065811
B-VP	O	1719.3	26125	0.039782
B-VP	B-ADJP	1039.3	26125	0.031169
B-VP	B-ADVP	814.3	26125	0.025428
B-VP	B-SBAR	664.3	26125	0.011265

Figure 8. Example data of transition probabilities of chunks

Chunk tag u_i	POS tag t_i	adj. count(u_i u_{i+1})	count(u_i)	P(t_i | u_i)
B-NP	at	19307.3	67503	0.286021
B-NP	nn	7555.3	67503	0.111925
B-NP	np	7541.3	67503	0.111718
B-NP	jj	5587.3	67503	0.082771
B-NP	nns	4473.3	67503	0.066268
B-NP	cd	2836.3	67503	0.042017
B-NP	pp$	2261.3	67503	0.033499
B-NP	pps	2080.3	67503	0.030818
B-NP	ppss	1620.3	67503	0.024003
I-NP	nn	34671.3	77683	0.446318
I-NP	nns	13143.3	77683	0.169191
I-NP	jj	8247.3	77683	0.106166
I-NP	np	7250.3	77683	0.093332
I-NP	cd	4727.3	77683	0.060854
I-NP	cc	1727.3	77683	0.022235
I-NP	vbg	1147.3	77683	0.014769
I-NP	vbn	940.3	77683	0.012104
I-NP	ap	862.3	77683	0.011100

Figure 9. Example data of emission probabilities of chunks

Using the chunk and clause information, we proceeded to extract a list of collocation types from the monolingual British National Corpus. We mainly used this list to identify collocation instances in SMEC. Finally, we applied the Competitive Linking Algorithm to SMEC to obtain word alignment results. We then applied the results of word alignment to extract the matching translations of the noun and verb collocates. The collocation extraction program produced a much larger set of collocation candidates than could be obtained from BNC. The corpus consists of over 100 million words in about 5 million sentences. After filtering out incomplete sentences, we obtained around 4 million sentences for use in extracting valid English collocations. After implementing our proposed method as described in Sections 2.2 and 2.3, we obtained over half a million collocation types of the forms VN, VPN, and VNP. We were able to identify over 30,000 collocation instances in SMEC. Figures 10 and 11 show some examples in BNC.

Type	Collocation types in the British Nation Corpus (BNC)	Collocation instances in the Sinorama Parallel Corpus (SPC)
VN	631,638	26,315
VPN	15,394	3,457
VNP	14,008	4,406

Figure 10. The results for collocation types extracted from the BNC and SMEC

VN type	Example
Exert influence	That means they would already be exerting their influence by the time the microwave background was born.
Exercise influence	The Davies brothers, Adrian (who scored 14 points) and Graham (four), exercised an important creative influence on Cambridge fortunes while their flankers Holmes and Pool-Jones were full of fire and tenacity in the loose.
Wield influence	Fortunately, George V had worked well with his father and knew the nature of the current political trends, but he did not wield the same influence internationally as his esteemed father.
Extend influence	The CAB extended its influence into the non-government sector, funding research by the Cathedral Advisory Commission and the Royal Society for the Protection of Birds.
Diminish influence	To break up the Union now would diminish our influence for good in the world, just at the time when it is most needed.
Gain influence	In general, women have not benefited much in the job market from capitalist industrialization nor have they gained much influence in society outside the family through political channels.
Counteract influence	To try and counteract the influence of the extremists, the moderate wing of the party launched a Labour Solidarity Campaign in 1981.
Reduce influence	Whether the curbs on police investigation will reduce police influence on the outcome of the criminal process is not easy to determine.

Figure 11. Examples of collocation instances extracted from SMEC

With the collocation types and instances extracted from the corpus, we built an on-line collocation reference tool called TANGO to support searching for collocations and translations of a given word.

Figure 12. TANGO, a web-based bilingual collocation tool

TANGO accepts a query word in English and a collocation type, and returns a list of collocation types and examples. Figure 12 shows a screen returned for a query for VN collocations of "influence." One instance for each collocation type is shown first. All instances can be shown on demand. Besides showing bilingual collocation extractions, TANGO also color codes the translation counterparts of the collocation instances. This informative, bilingual reference tool has been used in language learning classes and by professional translators. Initial responses have been quite positive, indicating that this new tool is very useful for EFL learners and translators.

To assess the quality of the extracted bilingual collocations, we randomly selected 100 sentences with extracted bilingual collocations from SMEC for manual evaluation. Many of these sentence had more than one collocation, 50 we evaluate each collocation individually. Students majoring in English assessed each bilingual collocation in the context of the corresponding pair of sentences. The evaluation process involved judging the validity of translation of the collocation. There were three levels of validity: satisfactory translation, approximate translation (partial matching), and unacceptable translation. Figure 13 shows examples for each level of validation. For the purpose of this research, satisfactory translations and approximate translations were considered useful. Therefore, we determined the percentages of bilingual collocations that fell into these two categories. As indicated in Figure 14, the average precision rate for the extracted bilingual collocations was about 90% for satisfactory translations and approximate translations.

Level of quality	English sentences	Chinese sentences
satisfactory translation	Thus when Chinpao Shan put out its advertisement last year, looking for new people to develop its related **enterprises**, the notice frankly stated "Southern Taiwanese preferred.	去年，金寶山在發展**關係企業**徵招新人的廣告上，就坦白指明「本省籍南部人優先」。
approximant translation	Ah-ying relates that "Teacher Chang" friendly and easy-going, is always there to answer her **questions**. She even goes to him for answers when her friends have legal questions.	阿英表示，「張老師」親切隨和，只要有不懂的事，都去問老師，就連朋友有法律上的**問題**，也去請教他。
unacceptable translation	Said one observer, "If I can speak bluntly, the mainlanders are robbing graves of their treasures and smuggling them away, and the situation is bad. In reality, though, it is Taiwan that is behind it all committing the **crime**.	「說得不好聽，大陸近年來盜墓、文物走私情形嚴重，台灣其實是背後的劊子手！」有人這樣**認爲**。

Figure 13. Three levels of quality of the extracted translation memory

Type	% of satisfactory translations	% of satisfactory and approximate translations*
VN	73	90
VPN	66	89
VNP	78	89

Figure 14. Evaluation of bilingual collocations extracted from SMEC

4. Discussion and limitation

Collocation is an important part of translation task yet it has long been neglected. Traditional machine translation tends to translate input texts word by word, which easily leads to literal translations. Therefore, even with abundant vocabulary, dictionary and grammar rule-based model systems fail to generate fluent translations in a target language. For example, due to its lack of collocational knowledge, a machine translation system may recognize “take” as “na” (i.e., take away) and “medicine” as “yao” (i.e., medicine) in Chinese, respectively. Thus, systems are inclined to literally translate “take medicine” as “na yao” (i.e., "take away the medicine" in Chinese), resulting in an odd translation or mistranslation. We suggest that machine translation systems should take collocational translation memory into consideration to improve the translation quality.

Due to the limitations of the word-alignment technique, our method may incorrectly recognize some matching translations. We need better word-alignment to align translations more correctly. Moreover, expansion of the bilingual corpora would also increase the precision achieved in retrieving collocational translation memory. This would enable us to obtain high enough counts for each collocate (i.e., verbs and nouns in VN collocations) in the target language so as to increase the confidence level of the LLR statistics, which in turn would eliminate the anomalous collocational translation memory.

5. Conclusion

In the field of machine translation, Example-Based Machine Translation (EBMT) exploits existing translations in the hope of producing better quality translations. However, collocational translation has always been neglected and is hard to deal with. We have proposed the use of collocational translation memory to develop a better translation method that can solve some problems resulting from literal translation. Encouraged by the satisfactory precision rates in collocation and translation extraction obtained in this study, we hope that collocational translation memory can be further applied in machine translation, cross language information retrieval, computer assisted language learning, and other NLP applications.

Acknowledgements

This work was part of the "CANDLE" project funded by the National Science Council of Taiwan (NSC-2213-E-238-015 and NSC92-2524-S007-002). Further information about CANDLE is available at http://candle.cs.nthu.edu.tw/.

References

Andriamanankasina, T., K. Araki, and T. Tochinai, "Example-Based Machine Translation of Parts-Of-Speech Tagged Sentences by Recursive Division," In *Proceedings of MT SUMMIT VII,* Singapore, 1999.

Brown, R. D., "Automated Generalization of Translation Examples," In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING-2000),* Saarbrücken, Germany, August 2000, pp. 125-131.

Carl, M., "Inducing Translation Templates for Example-Based Machine Translation," In *Proceedings of MT Summit VII*, 1999.

CoNLL yearly meeting of the SIGNLL, the Special Interest Group on Natural Language Learning of the Association for Computational Linguistics. The shared task of text chunking in CoNLL-2000 is available at http://cnts.uia.ac.be/conll2000/.

Echizen-ya, H., K. Araki, Y. Momouchi, and K. Tochinai, "Effectiveness of Automatic Extraction of Bilingual Collocations Using Recursive Chain-Link-Type Learning," *The 9th Machine Translation Summit*, 2003, pp.1 02-109.

Gao, J., J. Nie, H. He, W. Chen, and M. Zhou,"Resolving Query Translation Ambiguity Using a Decaying Co-occurrence Model and Syntactic Dependence Relations," *The 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 2000, pp.183 -190.

Kitano, H., "A Comprehensive and PracticalModel of Memory-Based Machine Translation," In *Proceedings of IJCAI-93*, 1993, pp. 1276-1282.

Kupiec, J., "An algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora," In *Proceedings of the 31th Annual Meeting of Association for Computational Linguistics*, 1993, pp. 17-22.

Lin, D., "Extracting Collocation from Text Corpora," *First Workshop on Computational Terminology*, 1998, pp. 57-63.

Lü, Y., and M. Zhou, "Collocation Translation Acquisition Using Monolingual Corpora," *Association for Computational Linguistics* 2004, pp. 167-174.

Melamed, I. D., "A Word-to-Word Model of Translational Equivalence," In *Proceedings of the Association for Computational Linguistics* 1997*, Madrid Spain,* 1997, pp. 490-497.

Nagao, M., "A Framework of a Mechanical Translation between Japanese and English by Analogy Principle," *Artificial and Human Intelligence*, A. Elithorn and R. Banerji (eds.) North-Holland, 1984, pp. 173-180.

Pearce, D., "Synonymy in Collocation Extraction," In *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations, CMU*, 2001.

Seretan, V., L. Nerima, and E. Wehrli, "Extraction of Multi-Word Collocations Using Syntactic Bigram Composition," *International Conference on Recent Advances in NLP*, 2003, pp. 424-431.

Manning, C.D., and H. Schutze, "Foundations of Statistical Natural Language Processing" *MIT Press, Cambridge, Mass,* 1999.

Smadja, F., "Retrieving Collocations from Text: Xtract," *Computational Linguistics*, 1993, 19(1), pp.143-177.

Smadja, F., K. R. Mckeown, and V. Hatzivassiloglou, "Translation Collocations for Bilingual Lexicons: a Statistical Approach," *Computational Linguistics*, 22, 1996, pp.1-38.

Wu, H., and M. Zhou, "Synonymous Collocation Extraction Using Translation Information," *The 4Jth annual conference of the Association for Computational Linguistics*, 2003, pp. 120-127

Related web pages:

Deja–Vu (http://www.atril.com/).

TOTALrecall (http://candle.cs.nthu.edu.tw/Counter/Counter.asp?funcID=1).

Transit (http://www.star-group.net/eng/software/sprachtech/transit.html).

TransSearch (http://www.tsrali.com/).

Computational Linguistics and Chinese Language Processing
Vol. 10, No. 3, September 2005, pp. 347-362

Detecting Emotions in Mandarin Speech

Tsang-Long Pao*, Yu-Te Chen*, Jun-Heng Yeh* and Wen-Yuan Liao*

Abstract

The importance of automatically recognizing emotions in human speech has grown with the increasing role of spoken language interfaces in human-computer interaction applications. In this paper, a Mandarin speech based emotion classification method is presented. Five primary human emotions, including anger, boredom, happiness, neutral and sadness, are investigated. Combining different feature streams to obtain a more accurate result is a well-known statistical technique. For speech emotion recognition, we combined 16 LPC coefficients, 12 LPCC components, 16 LFPC components, 16 PLP coefficients, 20 MFCC components and jitter as the basic features to form the feature vector. Two corpora were employed. The recognizer presented in this paper is based on three classification techniques: LDA, K-NN and HMMs. Results show that the selected features are robust and effective for the emotion recognition in the valence and arousal dimensions of the two corpora. Using the HMMs emotion classification method, an average accuracy of 88.7% was achieved.

Keywords: Mandarin, emotion recognition, LPC, LFPC, PLP, MFCC

1. Introduction

Research on understanding and modeling human emotions, a topic that has been predominantly dealt with in the fields of psychology and linguistics, is attracting increasing attention within the engineering community. A major motivation comes from the need to improve both the naturalness and efficiency of spoken language human-machine interfaces. Researching emotions, however, is extremely challenging for several reasons. One of the main difficulties results from the fact that it is difficult to define what emotion means in a precise way. Various explanations of emotions given by scholars are summarized in [Kleinginna *et al.* 1981]. Research on the cognitive component focuses on understanding the environmental and attended situations that give rise to emotions; research on the physical components emphasizes the physiological response that co-occurs with an emotion or rapidly follows it. In short,

* Department of Computer Science and Engineering, Tatung University, 40 ChungShan N. Rd., 3rd Sec, Taipei 104, Taiwan, R.O.C, Tel: +886-2-2592-5252 Ext. 2212, Fax: +886-2-2592-5252 Ext. 2288
E-mail: tlpao@ttu.edu.tw; {d8906005, d9306002, d8906004}@ms2.ttu.edu.tw

[Received January 31, 2005; Revised July 15, 2005; Accepted July 19, 2005]

emotions can be considered as communication with oneself and others [Kleinginna *et al.* 1981].

Traditionally, emotions are classified into two main categories: primary (basic) and secondary (derived) emotions [Murray *et al.* 1993]. Primary or basic emotions generally can be experienced by all social mammals (e.g., humans, monkeys, dogs and whales) and have particular manifestations associated with them (e.g., vocal/facial expressions, behavioral tendencies and physiological patterns). Secondary or derived emotions are combinations of or derivations from primary emotions.

Emotional dimensionality is a simplified description of the basic properties of emotional states. According to the theory developed by Osgood, Suci and Tannenbaum [Osgood *et al.* 1957] and in subsequent psychological research [Mehrabian *et al.* 1974], the computing of emotions is conceptualized as three major dimensions of connotative meaning: arousal, valence and power. In general, the arousal and valence dimensions can be used to distinguish most basic emotions. The locations of emotions in the arousal-valence space are shown in Figure 1, which provides a representation that is both simple and capable of conforming to a wide range of emotional applications.

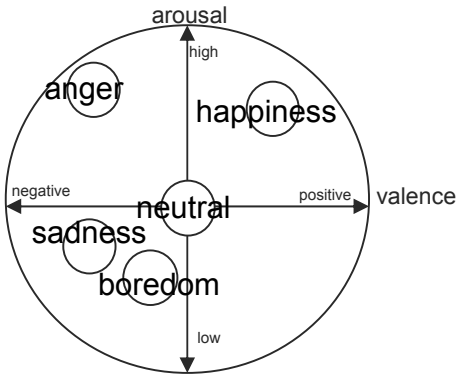

Figure 1. Graphic representation of the arousal-valence dimension of emotions[Osgood et al. 1957]

Numerous previous reports indicated that emotions could be detected by psychological cues [Cowie *et al.* 2000; Ekman 1999; Holzapfel *et al.* 2002; Inanoglu *et al.* 2005; Kleinginna *et al.* 1981; Kwon *et al.* 2003; Murray *et al.* 1993; Nwe *et al.* 2003; Park *et al.* 2002; Park *et al.* 2003; Pasechke *et al.* 2000; Picard 1997; Ververidis *et al.*2004]. Vocal cues are among the fundamental expressions of emotions, on a par with facial expressions [Cowie *et al.* 2000; Ekman 1999; Holzapfel *et al.* 2002; Kleinginna *et al.* 1981; Murray *et al.* 1993; Nwe *et al.*

2003; Park *et al.* 2002; Park *et al.* 2003; Pasechke *et al.* 2000; Ververidis *et al.* 2004]. All mammals can convey emotions by means of vocal cues. Humans are especially capable of expressing their feelings by crying, laughing, shouting and more subtle characteristics of speech.

In this paper, instead of modifying classifiers, we present an effective and robust set of vocal features for recognizing categories of emotions in Mandarin speech. The vocal characteristics of emotions are extracted from a Mandarin corpus. In order to surmount the inefficiency of conventional vocal features, such as pitch contour, loudness, speech rate and duration, for recognizing anger/happiness and boredom/sadness, we also adopt arousal and valence correlated characteristics to categorize emotions in emotional discrete categories. Several systematic experiments are presented. The characteristics of the extracted features are not only facile, but also discriminative.

The rest of this paper is organized as follows. In Section 2, two testing corpora are addressed. In Section 3, the details of the proposed system are presented. Experiments conducted to assess the performance of the proposed system are presented in Section 4 together with analysis of the results of the experiments. Concluding remarks are given in Section 5.

2. The Testing Corpora

An emotional speech database, Corpus I, was specifically designed and set up for emotion classification studies. The database includes short utterances portraying the five primary emotions, namely, anger, boredom, happiness, neutral and sadness. In the course of selecting emotional sentences, two aspects were taken into account. First, the sentences did not have any emotional tendency. Second, the sentences could involve all kinds of emotions. Non-professional speakers were selected to avoid exaggerated expression. Twelve native Mandarin language speakers (7 females and 5 males) were asked to generate the emotional utterances. The recording was done in a quiet environment using a mouthpiece microphone at a sampling rate of 8 kHz.

All of the native speakers were asked to speak each sentence with the five chosen emotions, resulting in 1,200 sentences. We first eliminated sentences that suffered from excessive noise. Then a subjective assessment of the emotion speech corpus by human audiences was carried out. The purpose of the subjective classification was to eliminate ambiguous emotion utterances. Finally, 558 utterances with over 80% human judgment accuracy were selected and are summarized in Table 1. In this study, utterances in Mandarin were used due to the immediate availability of native speakers of the language. It is easier for speakers to express emotions in their native language than in a foreign language. In order to accommodate the computing time requirement and bandwidth limitation of the practical

recognition application, e.g., the call center system [Yacoub *et al.* 2003], a sampling rate of 8 kHz was used. Another corpus, Corpus II, was recorded by Cheng [Cheng 2002]. Two professional Mandarin speakers were employed to generate 503 utterances with five emotions as shown in Table 2. The sampling rate was down-sampled to 8 kHz.

Table 1. Utterances for Corpus I

Emotion \ Sex	Female	Male	Total
Anger	75	76	151
Boredom	37	46	83
Happiness	56	40	96
Neutral	58	58	116
Sadness	54	58	112
Total	280	278	558

Table 2. Utterances for Corpus II

Emotion \ Sex	Female	Male	Total
Anger	36	72	108
Boredom	72	72	144
Happiness	36	36	72
Neutral	36	36	72
Sadness	72	35	107
Total	252	251	503

Utterances can be divided into two sets: one set for training and one set for testing. In this way, several different models, all trained with the training set, can be compared based on the test set. This is the basic form of cross-validation. A better method, which is intended to avoid possible bias introduced by relying on any one particular division into test and train components, is to partition the original set in several different ways and then compute an average score over the different partitions. An extreme variant of this is to split the p patterns into a training set of size p-1 and a test of size 1, and average the squared error on the left-out pattern over the p possible ways of obtaining such a partition. This is called leave-one-out (LOO) cross-validation. The advantage here is that all the data can be used for training; none have to be held back in a separate test set.

3. Emotion Recognition Method

The proposed emotion recognition method has three main stages: feature extraction, feature vector quantization and classification. Base features and their statistics are computed in the feature extraction stage. Feature components are quantized into a feature vector in the feature

quantization stage. Classification is done by using various classifiers based on dynamic models or discriminative models.

3.1 Emotion Feature Selection

Determining emotion features is a crucial issue in emotion recognizer design. All selected features have to carry sufficient information about transmitted emotions. However, they also need to fit the chosen model by means of classification algorithms. Important research was done by Murray and Arnott [Murray *et al.* 1993], whose results particularized several notable acoustic attributes for detecting primary emotions. Table 3 summarizes the vocal effects most commonly associated with the five primary emotions [Murray *et al.* 1993]. Classification of emotional states based on prosody and voice quality requires classifying the connections between acoustic features in speech and emotions. Specifically, we need to find suitable features that can be extracted and modeled for use in recognition. This also implies that the human voice carries abundant information about the emotional state of a speaker.

Table 3. Emotions and speech relations [Murray et al. 1993]

	Anger	**Happiness**	**Sadness**	**Fear**	**Disgust**
Speech Rate	Slightly faster	Faster or slower	Slightly slower	Much faster	Very much faster
Pitch Average	Very much higher	Much higher	Slightly lower	Very much higher	Very much lower
Pitch Range	Much wider	Much wider	Slightly narrower	Much wider	Slightly wider
Intensity	Higher	Higher	Lower	Normal	Lower
Voice Quality	Breathy, chest	Breathy, blaring tone	Resonant	Irregular voicing	Grumble chest tone
Pitch changes	Abrupt on stressed	Smooth, upward inflections	Downward inflections	Normal	Wide, downward terminal inflects
Articulation	Tense	Normal	Slurring	Precise	Normal

A variety of acoustic features have also been explored. For example, Schuller *et al.* chose 20 pitch and energy related features [Schuller *et al.* 2003]. A speech corpus consisting of acted and spontaneous emotion utterances in German and English was described in detail. The accuracy in recognizing 7 discrete emotions (anger, disgust, fear, surprise, joy, neutral and sad) exceeded 77.8%. Park *et al*. used pitch, formant, intensity, speech rate and energy related features to classify neutral, anger, laugh and surprise [Park *et al.*2002]. The recognition rate was about 40% for a 40-sentence corpus. Yacoub *et al*. extracted 37 fundamental frequency, energy and audible duration features for recognizing sadness, boredom, happiness and anger in a corpus recorded by eight professional actors [Yacoub *et al.*2003]. The overall accuracy

was only about 50%, but these features successfully separated hot anger from other basic emotions. Tato *et al.* extracted prosodic features, derived from pitch, loudness, duration and quality features [Tato *et al.*2002], from a 400-utterance database. The significant results of emotion recognition were the speaker-independent case and three clusters (high = anger/happy, neutral, low = sad/bored). However, the accuracy in recognizing five emotions was only 42.6%. Kwon *et al.* selected pitch, log energy, formant, band energies and Mel frequency spectral coefficients (MFCC) as base features, and added velocity/acceleration of pitch to form feature streams [Kwon *et al.2003*]. The average classification accuracy achieved was 40.8% in a SONY AIBO database. Nwe *et al.* adopted the short time log frequency power coefficients (LFPC) along with MFCC as emotion speech features to recognize 6 emotions in a 60-utterance corpus produced by 12 speakers [Nwe *et al.*2003]. Results showed that the proposed system yielded an average accuracy of 78%. In [Le *et al.* 2004], the authors proposed a method using MFCC coefficients and a simple but efficient classifying method, Vector Quantization, for performing speaker-dependent emotion recognition. Various speech features, namely, energy, pitch, zero crossing, phonetic rate, LPC and their derivatives, were also tested and combined with MFCC coefficients. The average recognition accuracy achieved was about 70%. In [Chuang *et al.* 2004], Chuang and Wu presented an approach to emotion recognition from speech signals and textual content using PCA and SVM, and achieved 81.49% average accuracy using an extra corpus collected from the same broadcast drama.

According to the experimental results stated above, some simple prosodic features, such as duration, loudness, can not consistently distinguish all primary emotions. Furthermore, the prosodic features of females and males are obviously intrinsic in speech. The simple speech energy feature calculation method is also unconformable to human auricular perception.

Figure 2 shows a block diagram of the feature extraction process. In the pre-processing procedure, locating the endpoints of the input speech signal is done first. The speech signal is high-pass filtered to emphasize the important high frequency components. Then the speech frame is partitioned into frames consisting of 256 samples each. Each frame overlaps with the adjacent frames by 128 samples. The next step is to apply the Hamming window to each individual frame to minimize the signal discontinuities at the beginning and end of each frame. Each windowed speech frame is then converted into several types of parametric representations for further analysis and recognition.

In order to find a suitable combination of extracted features, we used the regression selection method to determine beneficial features from among more than 200 speech features. Ten candidates were selected: LPC, LPCC, MFCC, Delta-MFCC, Delta-Delta-MFCC, PLP, RastaPLP, LFPC, jitter and shimmer. Then the feature vector of each frame of a sentence from corpus I was calculated. The recognition rate in each step was calculated using the LOO cross-validation method with the K-NN (K=3) classifier.

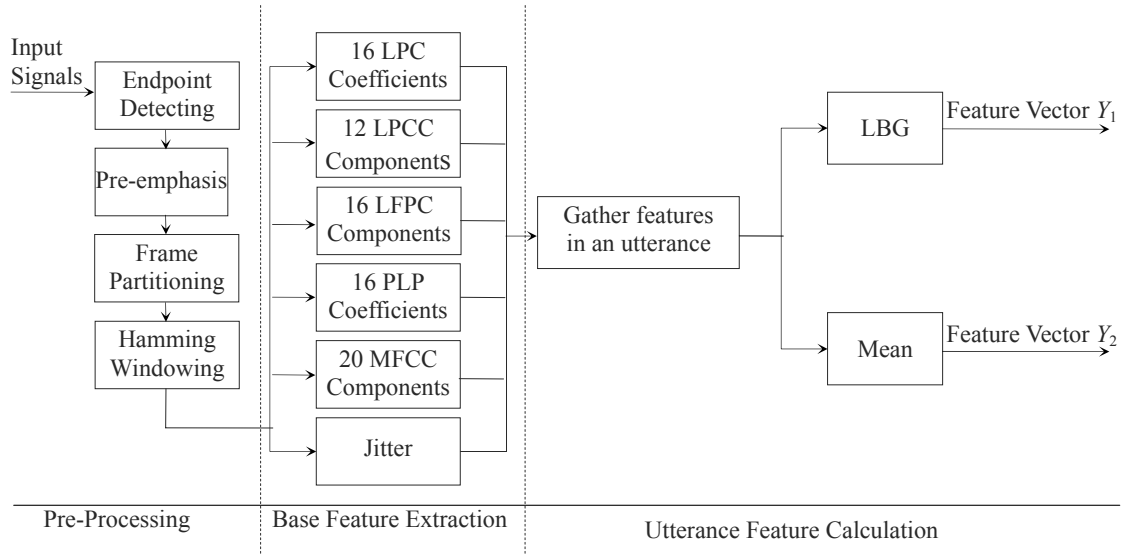

Figure 2. Block diagram of the feature extraction module

Table 4 shows the recognition rate of the first 10 candidates. The highest recognition rate was found to be 83.91% using the forward selection procedure shown in Table 6. In this procedure, the recognition rate grows or declines according to the effectiveness of feature combining. Tables 4-6 list the results of forward selection with 1, 2 and 6 features. Based on these experimental results, we selected six features, which were LPCC, MFCC, LFPC, jitter, PLP and LPC, as a beneficial feature combination for speech emotion recognition.

Table 4. The recognition rate with single feature

Feature	**Accuracy (%)**
LPCC	68.68
MFCC	68.21
LPC	68.20
PLP	65.59
RastaPLP	65.23
D-MFCC	60.59
LFPC	58.42
Shimmer	53.05
D-D-MFCC	50.18
Jitter	34.77

Table 5. The recognition rate with two feature sets

Feature		Accuracy (%)
LPCC	MFCC	68.97
	D-MFCC	67.38
	LPC	66.52
	PLP	66.52
	LFPC	66.52
	RastaPLP	66.16
	D-D-MFCC	60.06
	Jitter	54.33
	Shimmer	42.86

Table 6. The recognition rate with six feature sets

Feature		Accuracy (%)
LPCC MFCC LFPC Jitter PLP	LPC	83.91
	RastaPLP	83.91
	D-MFCC	83.19
	D-D-MFCC	83.19
	Shimmer	79.40

In the base feature extraction procedure, we selected six types of features, which were 16 Linear predictive coding (LPC) coefficients, 12 linear prediction cepstral coefficients (LPCC), 16 log frequency power coefficients (LFPC), 16 perceptual linear prediction (PLP) coefficients, 20 Mel-frequency cepstral coefficients (MFCC) and jitter extracted from each frame. This added up to a feature vector consisting of 81 parameters. LPC provides an accurate and economical representation of the envelope of the short-time power spectrum of speech [Kaiser 2002]. For speech emotion recognition, LPCC and MFCC are popular choices as they represent the phonetic content of speech and convey information about short time energy migration in the frequency domain [Ata 1997; Davis *et al.* 1980]. LFPC is calculated using a log frequency filter bank, which can be regarded as a model that shows the varying auditory resolving power of the human ear for various frequencies [Nwe *et al.*2003]. The combination of the discrete Fourier transform (DFT) and LPC technique is called PLP [Hermansky 1990]. PLP analysis is computationally efficient and permits a compact representation. Perturbations in the pitch period are called jitter. Such perturbations occur naturally during continuous speech.

3.2 Feature Vector Quantization

Each feature vector consists of 81 parameters, which requires intensive computation when classification is performed. To compress the data in order to accelerate the classification process, vector quantization is performed. All the vectors of a frame falling into a particular cluster are coded with the vector representing that cluster. The vector is assigned the codeword c_n^*, according to the best matching codebook cluster. An experiment was conducted with different numbers of centroids obtained using the Linde-Buzo-Gray(LBG) K-means algorithm [Linde *et al.* 1980]. It was found that the effectiveness per centroid diminished significantly when the size exceeded 16. In this study, we took 16 as the number of LBG centroids in all of the experiments. For each utterance with *N* frames, the feature vector Y_1 with 16*81 parameters was then obtained in the form

$$Y_1 = [c_1^* c_2^* ... c_N^*]. \tag{1}$$

Another simple vector quantization method used the mean of the feature parameters corresponding to each frame in one utterance to form a feature vector Y_2 with 81 parameters as follows:

$$Y_2 = [p_1 p_2 ... p_{81}], \tag{2}$$

where p_i is the mean value of the *i*th parameter of all frames.

3.3 Classifiers

Three different classifiers, linear discriminate analysis (LDA), the k-nearest neighbor (K-NN) decision rule, and Hidden Markov models (HMMs), were used to train and test these two testing emotion corpora with the extracted features from Corpus I. In the K-NN decision rule, there are three nearest samples that are closest to the testing sample. In HMMs, the state transition probabilities and output symbol probabilities are uniformly initialized. Our experimental results show that the 4-state discrete ergodic HMM achieved the best performance compared with the left-right structure.

4. Experimental Results

The selected features were quantized using the LBG algorithm to form the vector Y_1 and quantized using the mean method to form vector Y_2. Then the feature vectors were trained and tested with all three classifiers, which were LDA, K-NN and HMMs. All of the experimental results were validated using the LOO cross-validation method.

4.1 The Experimental Results Obtained with the Conventional Prosodic Features

In [Kwon *et al.* 2003], Kwon *et al.* drew a two-dimensional plot of 59 features ranked by means of forward selection and backward elimination. Features near the origin were considered to be more important. By imitating the ranking features method as in [Kwon *et al.* 2003], we could rank the speech features extracted from Corpus I through forward selection and backward elimination as shown in Figure 3. Our experimental results and the Kwon's both show that the pitch and energy related features are the most important components for emotion speech recognition in both Mandarin and English. We selected the first 15 features proposed in [Kwon *et al.* 2003] from Corpus I to examine the efficiency and stability of the conventional emotion speech features. The first 15 features were pitch, log energy, F1, F2, F3, 5 filter bank energies, 2 MFCCs, delta pitch, acceleration of pitch and 2 acceleration MFCCs. Then the feature vector Y_2 and K-NN were used.

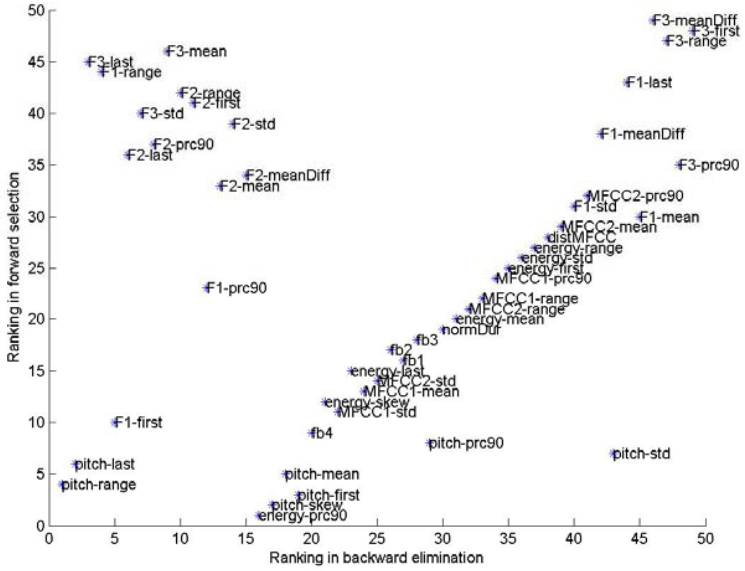

Figure. 3 Ranking of conventional speech features

The confusion matrix that employs conventional emotion speech features is shown in Table 7. The overall average accuracy achieved for the five primary emotions was 53.2%. Similar to most of the previous researches, the pitch and energy related features extracted from the time domain had difficulty distinguishing anger and happiness. The reason is that anger and happiness are close to each other in pitch and energy. Hence, the classifiers often confuse one with the other. This also applies to boredom and sadness.

Table 7. Experimental results obtained using conventional prosodic features

Accuracy (%)	Anger	Boredom	Happiness	Neutral	Sadness
Anger	**59.5**	1.1	32.4	4.4	2.6
Boredom	0	**46.8**	1.1	20.4	31.7
Happiness	32.4	2.5	**58.7**	4.2	2.2
Neutral	9.4	7.7	8.7	**52.1**	22.1
Sadness	1.7	29.4	2.4	17.6	**48.9**

4.2 Experimental Results of Valence Emotions Recognition

The prosodic features related to pitch and energy failed to distinguish the valence emotions. The selected features discussed in Section 3.1 were quantized into feature vector Y_1 and mean feature vector Y_2. The feature vectors from Corpus I were then trained and tested using three different classifiers, the LDA, K-NN and HMMs. All the experimental results were validated using the LOO cross-validation method. According to the experimental results shown in Tables 8 and 9, the three recognizers were undoubtedly able to separate anger and happiness, which most previous emotion speech recognizers usually confuse.

The pairwise emotions, anger and happiness, are considered to be close to each other in the arousal dimension, having similar prosody and amplitude. So do boredom and sadness. The conventional speech emotion recognition method suffers from ineffectiveness and instability in emotion recognition, especially for emotions in the same arousal dimension. On the other hand, using the selected features in the proposed system solves this problem and results in a high recognition rate. The selected features are not only suitable for various classifiers but also effective for speech emotion recognition.

Table 8. Experimental results of anger and happiness recognition

Accuracy (%)	LDA		K-NN		HMMs	
	Y_1	Y_2	Y_1	Y_2	Y_1	Y_2
Anger	93.1	93.4	93.7	91.6	93.9	92.6
Happiness	87.7	91.2	90.4	92.8	91.2	93.5
Average	90.4	92.3	92.0	92.2	92.5	93.0

Table 9. Experimental results of boredom and sadness recognition

Accuracy (%)	LDA		K-NN		HMMs	
	Y_1	Y_2	Y_1	Y_2	Y_1	Y_2
Boredom	89.5	90.5	89.7	92.1	90.5	94.3
Sadness	92.2	87.6	93.5	90.4	93.2	90.9
Average	90.8	89.0	91.6	91.0	91.8	92.6

4.3 Experimental Results for Corpus I and Corpus II

Tables 10 and 11 show the accuracy achieved in classifying the five primary emotions using various classifiers and two feature vector quantization methods applied to Corpus I and II. The various classifiers differ in ability and properties. Hence we achieved various recognition accuracy results with the different classifiers and quantization methods.

Table 10. Experimental results for five emotion categories in Corpus I

Accuracy (%)	LDA		K-NN		HMMs	
	Y_1	Y_2	Y_1	Y_2	Y_1	Y_2
Anger	81.5	80.4	82.3	84.8	86.4	86.7
Boredom	80.3	79.8	84.9	82.3	89.1	88.4
Happiness	76.5	72.3	79.5	82.1	82.3	83.6
Neutral	78.4	80.5	80.4	81.2	84.5	90.5
Sadness	82.5	81.3	91.2	89.1	92.4	92.3
Average	79.8	78.8	83.6	83.9	86.9	88.3

Table 11. Experimental results for emotion categories in Corpus II

Accuracy (%)	LDA		K-NN		HMMs	
	Y_1	Y_2	Y_1	Y_2	Y_1	Y_2
Anger	82.4	76.2	83.2	84.5	90.2	91.4
Boredom	78.9	80.2	81.5	80.9	84.3	86.7
Happiness	81.4	77.8	86.4	82.5	87.5	88.1
Neutral	76.5	79.8	84.1	83.2	90.3	86.0
Sadness	80.3	76.5	86.0	87.5	89.5	91.5
Average	79.9	78.1	84.2	83.7	88.3	88.7

According to the experimental results shown in Tables 10 and 11, the overall accuracy rates achieved for the five primary emotions, namely, anger, boredom, happiness, neutral and sadness, were about the same. In addition, the accuracy rates of the two feature quantization methods were quite close to each other when used under the same conditions. This shows that the set of selected speech features is stable and suitable for recognizing the five primary emotions, using various classifiers with different feature quantization methods. Based on the high recognition accuracy rates achieved for Corpus I and Corpus II, the selected features can be efficiently used to classify the five primary emotions of the arousal and the valence degree simultaneously.

Two different corpora were used to validate the robustness and effectiveness of the selected features. From the experimental results shown in Tables 10 and 11, the overall recognition rates obtained for both corpora are similar.

5. Conclusion

Dealing with the emotions of speaker is one of the challenges for speech processing technologies. Whereas the research on automated recognition of emotions in facial expressions has been quite extensive, that focusing on speech modality, both for automated production and recognition by machines, has been active only in recent years and has mostly focused on English. Possible applications include intelligent speech-based customer information systems, human oriented human-computer interaction GUIs, interactive movies, intelligent toys and games, situated computer-assisted speech training systems and supported medical instruments.

The selection of a feature set is a critical issue for all recognition systems. In the conventional approach to emotion classification of speech signals, the features typically employed are the fundamental frequency, energy contour, duration of silence and voice quality. However, previous proposed recognition methods employing these features perform poorly in recognizing valence emotions. In addition, these features, when applied to different corpora, obtain different recognition results with the same recognizer.

In this study, we combined 16 LPC coefficients, 12 LPCC components, 16 LFPC components, 16 PLP coefficients, 20 MFCC components and jitter as features, and used LDA, K-NN and HMMs as the classifiers. The emotions were classified into five human primary categories: anger, boredom, happiness, neutral and sadness. Two Mandarin corpora, one consisting of 558 emotional utterances made by 12 native speakers and the other consisting of 503 emotional utterances made by 2 professional speakers, were used to train and test the proposed recognition system. Results obtained show that the proposed system yielded top recognition rates of 88.3% for Corpus I and 88.7% for Corpus II.

According to the experimental outcomes, we attained high recognition rates in distinguishing anger/happy and bored/sad emotions, which have similar prosody and amplitude. The proposed method can solve the problem of recognizing valence emotions using a set of extracted features. Moreover, the recognition accuracy results for Corpus I and Corpus II show that the selected speech features are suitable and effective for the speech emotion recognition with different corpora.

Further improvement and expansion may be achieved according to the following suggestions: The set of the most efficient features for emotion recognition is still vague. A possible approach to extracting non-textual information to identify emotional states in speech is to apply all known feature extraction methods. Thus, we may try to incorporate the information of different features into our system to improve the accuracy of emotion recognition. Recognizing emotion translation in real human communication is also a challenge. Thus, it will be worth while to determine the points where emotion transitions occur.

Acknowledgement

The authors are thankful to the National Science Council of Taiwan, R.O.C., for financial supports of this work (grant no. 93-2213-E-036-023 and 92-2213-E-036-021).

References

Ata, B.S., "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification," *Journal of the Acoustical Society of America*, 1(55), 1974, pp.1304-1312.

Cheng, P.Y., "Automated Recognition of Emotion in Mandarin," MD thesis, National Cheng Kung University, 2002.

Chuang, Z.J. and C.H. Wu, "Multi-Modal Emotion Recognition from Speech and Text," *International Journal of Computational Linguistics and Chinese Language Processing*, 9(2), 2004, pp.1-18.

Cowie, R., E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz and J.G. Taylor, "Emotion Recognition in Human-Computer Interaction," *IEEE Signal Processing Magazine*, 18 (1), 2000, pp.32-80.

Davis, S. and P. Mermelstein, "Comparison of Parametric Representations of Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics*, 28(4), 1980, pp.357-366.

Ekman, P., *Handbook of Cognition and Emotion*, John Wiley & Sons, New York, 1999.

Hermansky, H., "Perceptual Linear Predictive (PLP) Analysis of Speech," *Journal of the Acoustical Society of America*, 87(4), 1990, pp.1738-1752.

Holzapfel, H., C. Fügen, M. Denecke and A. Waibel, "Integrating Emotional Cues into a Framework for Dialogue Management," In *Proceedings of International Conference on Multimodal Interfaces*, 2002, Pennsylvania, USA, pp.141-148.

Inanoglu, Z. and R. Caneel, "Emotive Alert: HMM-Based Emotion Detection in Voicemail Messages," In *Proceedings of Intelligent User Interfaces*, 2005, San Diego, USA, pp.251-253.

Kaiser, J.F., *Discrete-Time Speech Signal Processing*, Prentice Hall, New Jersey, 2002.

Kleinginna , P.R. and A.M. Kleinginna, "A Categorized List of Emotion Definitions with Suggestions for a Consensual Definition," *Motivation and Emotion*, 5(4), 1981, pp.345-379.

Kwon, O.W., K. Chan, J. Hao and T.W. Lee , "Emotion Recognition by Speech Signals," In *Proceedings of Eurospeech*, 2003, Geneva, Switzerland, pp.125-128.

Le, X.H., G. Quenot and E. Castelli, "Recognizing Emotions for the Audio-Visual Document Indexing," In *Proceedings of the Ninth IEEE International Symposium on Computers and Communications*, 2004, Alexandria, Egypt, pp.580-584.

Linde, Y., A. Buzo and R.M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, 28(1), 1980, pp.84-95.

Mehrabian, A. and J. Russel, *An Approach to Environmental Psychology*, The MIT Press, Cambridge, 1974.

Murray, I. and J.L. Arnott, "Towards the Simulation of Emotion in Synthetic Speech: A Review of the Literature on Human Vocal Emotion," *Journal of the Acoustic Society of America*, 93(2), 1993, pp.1097-1108.

Nwe, T.L., S.W. Foo and L.C. De-Silva, "Speech Emotion Recognition Using Hidden Markov Models," *Speech Communication*, 41(4), 2003, pp.603-623.

Osgood, C.E., J.G. Suci and P.H. Tannenbaum, *The Measurement of Meaning*, The University of Illinois Press, Urbana, 1957.

Park, C.H., K.S. Heo, D.W. Lee, Y.H. Joo and K.B. Sim, "Emotion Recognition based on Frequency Analysis of Speech Signal," *International Journal of Fuzzy Logic and Intelligent Systems*, 2(2), 2002, pp.122-126.

Park, C.D. and K.B. Sim, "Emotion Recognition and Acoustic Analysis from Speech Signal," In *Proceedings of International Joint Conference on Neural Networks*, 2003, Portland, USA, pp.2594-2598.

Pasechke, A. and W.F. Sendlmeier, "Prosodic Characteristics of Emotional Speech: Measurements of Fundamental Frequency Movements," In *Proceedings of ISCA Workshop on Speech and Emotion*, 2000, Northern Ireland, pp.75-80.

Picard, R.W., *Affective Computing*, The MIT Press, Cambridge, 1997.

Schuller, B., G. Rigoll, and M. Lang, "Hidden Markov Model-based Speech Emotion Recognition," In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2003, Hong Kong, China, pp.401-405.

Tato, R.S., R. Kompe and J.M. Pardo., "Emotional Space Improves Emotion Recognition," In *Proceedings of International Conference on Spoken Language Processing*, 2002, Colorado, USA, pp.2029-2032.

Ververidis, D., C. Kotropoulos and I. Pitas, "Automatic Emotional Speech Classification," In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2004, Montreal, Canada, pp.593-596.

Yacoub, S., S. Simske, X. Lin and J. Burns, "Recognition of Emotions in Interactive Voice Response Systems," In *Proceedings of Eurospeech*, 2003, Geneva, Switzerland, pp.729-732.

Computational Linguistics and Chinese Language Processing
Vol. 10, No. 3, September 2005, pp. 363-380

Modeling Pronunciation Variation for Bi-Lingual Mandarin/Taiwanese Speech Recognition

Dau-Cheng Lyu [*,], Ren-Yuan Lyu [*], Yuang-Chin Chiang [+] and Chun-Nan Hsu [**]**

Abstract

In this paper, a bi-lingual large vocaburary speech recognition experiment based on the idea of modeling pronunciation variations is described. The two languages under study are Mandarin Chinese and Taiwanese (Min-nan). These two languages are basically mutually unintelligible, and they have many words with the same Chinese characters and the same meanings, although they are pronounced differently. Observing the bi-lingual corpus, we found five types of pronunciation variations for Chinese characters. A one-pass, three-layer recognizer was developed that includes a combination of bi-lingual acoustic models, an integrated pronunciation model, and a tree-structure based searching net. The recognizer's performance was evaluated under three different pronunciation models. The results showed that the character error rate with integrated pronunciation models was better than that with pronunciation models, using either the knowledge-based or the data-driven approach. The relative frequency ratio was also used as a measure to choose the best number of pronunciation variations for each Chinese character. Finally, the best character error rates in Mandarin and Taiwanese testing sets were found to be 16.2% and 15.0%, respectively, when the average number of pronunciations for one Chinese character was 3.9.

Keywords: Bi-lingual, One-pass ASR, Pronunciation Modeling

1. Introduction

Words can be pronounced in more than one ways according to a lexicon; i.e., they usually have multiple pronunciations. Words are also pronounced differently by different people, a

[*] Chang Gung University, Taiwan
E-mail: rylyu@mail.cgu.edu.tw
[+] National Tsing Hua University, Taiwan
[**] Academia Sinica, Taiwan
E-mail: {daucheng, chunnan}@iis.sinica.edu.tw

[Received February 22, 2005; Revised July 22, 2005; Accepted August 15, 2005]

phenomenon called "pronunciation variation." Pronunciation variation has been studied in the speech recognition field [Chen 1996; Cremelie 1996], and reports show that pronunciation variation can cause the performance of automatic speech recognizers to deteriorate if it is not well accounted for. A common approach to solving the pronunciation variation problem is to use pronunciation modeling; where multiple pronunciations are added to each lexeme in a lexicon in order to fit the acoustic data better.

A Chinese character is pronounced differently in different languages which use that Chinese character in their writing systems. The same character may or may not have the same meaning in such languages. For instance, the Chinese character "窗"(window) is pronounced "chuang^{11}" in Mandarin and "tang^{11}" in Taiwanese, and these are considered to be multiple pronunciations in a Mandarin/Taiwanese bi-lingual lexicon. "Chuang^{11}" is often mistakenly pronounced "cuang^{11}" (the un-retroflex of "chuang^{11}") by native Taiwanese speakers, who do not have un-retroflex consonants in their language. This is a common cause of pronunciation variation. In the case of English, which has a more complex vowel inventory than the Han language family, the words "ear" and "year" are difficult for Mandarin speakers to tell apart. In other words, pronunciation variation is a natural and unavoidable phenomenon in a multi-lingual environment.

In this world of people who are well-connected by various types of communication devices, multi-lingual communication is necessary, and multi-lingual speech recognition is a must. This paper focuses on Mandarin-Taiwanese bi-lingual large vocabulary speech recognition, and the framework studied here is applicable to other language combinations as well.

Studies on the pronunciation variation problem have focused on two basic approaches, which are based on acoustic modeling or pronunciation modeling. For acoustic modeling, reports [Jurafsky *et al.* 2001] show that the triphone model can well capture variation resulting from phone substitution or phone reduction; other reports [Liu *et al.* 2003; Kam *et al.* 2003] show that well-trained triphone acoustic models can handle partial change of the pronunciation variation which depends on the context.

In pronunciation modeling, entries in the pronunciation dictionary include alternative pronunciation variations and associated probabilities, determined through either knowledge-based or data-driven approaches [Kipp *et al*. 1996; Zeppenfeld *et al*. 1997; Wiseman *et. al*. 1998; Wester 2003; Polzin *et al*. 1998; Peters *et al*. 1998; Bacchiani *et al*. 1999; Singh *et al*. 2002; Kessens 2003; Strik 2003.]. With the knowledge-based approach, variation information is obtained from research reports or pronunciation dictionaries. Techniques for obtaining the probabilities of possible pronunciation variations of a word in the data-driven approach include training decision trees, training an artificial network, using entropy, using the maximum likelihood criterion, and using the calculated phone confusion

matrix [Cremelie and Martens 1998; Riley *et al*. 1999; Kam *et al*. 2003; Fukada *et al*. 1997; 1998; Yang *et al.* 2000; Holter *et al*. 1999; Torre *et al*. 1997]. Techniques that achieve higher scores are chosen to serve as pronunciation variation rules.

In addition to the pronunciation variation within a word, substantial variation occurs across word boundaries [Finke *et al.* 1997; Fukada *et al.* 1998; Kessens *et al.* 1999.]. Due to the mono-syllabic nature of Mandarin and Taiwanese, pronunciation variation is complex, and we can identify five types of variation: (1) one orthography with pronunciation variation; (2) colloquial/literate switching; (3) tone sandhi; (4) one orthography with multiple pronunciations; (5) one pronunciation with multiple orthography. The first three types of variation occur in mono-lingual environment, while the last two occur in bi-lingual environments. Details will be given in Section 3.

The goal of this study was to construct a Mandarin/Taiwanese bi-lingual large vocabulary speech recognizer. We implemented a one-pass recognizer based on a bi-lingual acoustic model, an integrated pronunciation model, and a word searching net with tree-structured nodes. Most of the state-of-the-art speech recognizers, for either Western or Oriental languages, are implemented with the one-pass search strategy [Odell 1994; Aubert 1999; Hagen 2001]. In the acoustic modeling, one phonemic inventory called ForPA (Formosa Phonetic Alphabet) is used to transcribe bi-lingual corpora. [Lyu *et al.* 2004] According to this inventory, the acoustic models for similar sounds across languages are shared. In addition, we use an algorithm based on a decision tree to cluster similar acoustic models by means of the maximum likelihood criterion. In the pronunciation modeling, we integrate knowledge-based and data-driven approaches. If only the knowledge-based approach is adopted, some variation in the speech corpus can not be covered at all, while if only the data-driven approach is employed, the variation for each new corpus has to be determined. However, the more variations for each word there are in the searching net, the more the recognition time and confusability will increase. To limit the number of pronunciation variations for each Chinese character, we adopt a score based on the relative frequency ratio and choose the best average number of pronunciation variations. Furthermore, the tree-structured net directly uses each Chinese character as a searching node, which is also a new trial in the ASR field of Chinese languages.

This paper is organized as follows. Section 2 states the problem. Section 3 represents the proposed framework, which includes acoustic modeling, pronunciation modeling, and a searching net. In section 4, we report experimental results and analyze three different pronunciation models using a bi-lingual testing set. The final section is a summary.

2. Problem Statements

In recent decades, most of the speech recognition research related to the Chinese (also called the "漢" Han) language family has focused on Mandarin speech [Lee 1998; Liao *et al*. 2000]. Relatively few studies have focused on other languages [Lyu *et al.* 2000; Gau *et al.* 2000]. In this paper, we consider two languages in this language family, i.e., Mandarin and Taiwanese, simultaneously within the same framework of speech recognition. In Taiwan, Mandarin Chinese is the official language, and Taiwanese is the mother tongue of about three quarters of the population. Quite a few people speak Mandarin with an accent that is strongly influenced by Taiwanese, and when they speak Taiwanese, they mix in words from Mandarin. It appears that people in Southern China do much the same. If successful, we expect that this framework will work well for other combinations of Chinese languages.

In the Mandarin Chinese speech recognition system, a typical syllable decoder is implemented by searching a 3-layer network consisting of an acoustic model layer, a lexical layer, and a grammar layer, as shown in Figure 1. After the optimal syllable sequence or the syllable lattice is determined by the decoder, a syllable-to-character converter is applied to handle the homonym issue for the final text output, as shown in Figure 2. This framework works well and has long been used by the speech communication community. To generalize the system so as to incorporate more than one language, a straightforward approach is to extend the system with more acoustic models, more entries in the pronunciation dictionary, and more paths in the searching net. However, this will lead to the following difficulties:

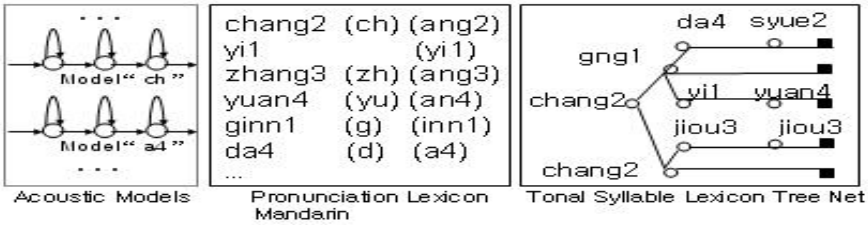

Figure 1. A 3-layer grammar searching net for syllable decoding

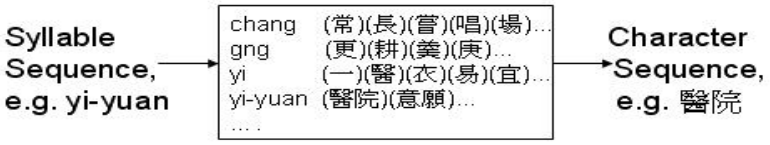

Figure 2. The syllable-to-character converter

1. In the case of multi-syllabic words such as "國家" (country), people rarely use Mandarin pronunciation for part of the word and Taiwanese pronunciation for the other part. It is, thus, impractical to generate all instances of all possible bi-lingual pronunciation variations of each

character in a word for a recognition network. Doing so will not only unnecessarily enlarge the searching space but also increase the time spent on decoding.

2. Generating multiple pronunciation lexicons efficiently is not a trivial task.

3. The language model for mixed languages is hard to estimate.

4. When new acoustic features like tones are added to the system, all 3 layers in syllable decoding and in the syllable-to-character converter should be modified. This also is not a trivial task.

3. Our Approach

Unlike some conventional approaches, which divide the recognition task into syllable decoding and character decoding, our proposed approach adopts a one-stage searching strategy, as shown in Figure 3, which decodes the acoustic feature sequence X directly to obtain the desired character sequence C*, no matter what languages are spoken. The decoding equation is, thus, as follows:

$$C^*(X) = \arg\max_{C} P(C \mid X). \tag{1}$$

In this framework, character decoding can be implemented by searching in a three-layer network composed of an acoustic model layer, a lexical layer, and a grammar layer, as shown in Figure 4. There are at least 2 critical differences between our framework and the conventional one. 1) In the lexicon layer, character-to-pronunciation mapping can easily incorporate multiple pronunciations caused by multiple languages, including Japanese, Korean, and even Vietnamese, which also use Chinese characters. 2) In the grammar layer, characters instead of syllables are used as nodes in the searching net. Under this ASR structure, we do not care which language the user speaks. No matter whether the language is Taiwanese, Mandarin or a mixture of them in one sentence, the ASR outputs the Chinese character only. This makes it language independent!

As in other multi-lingual researches [Young *et al*. 1997; Waibel 2000], determining how to efficiently and easily combine two languages in the acoustic and pronunciation models is very important. In the following two subsections, we will describe various approaches to integrating these two models in order to improve the recognition performance of ASR systems.

Figure 3. One-stage searching strategy for Chinese speech recognition

Figure 4. A unified 3-layer framework for multi-language Chinese speech recognition

3.1 Unified Bi-lingual Acoustic Modeling

It has been shown that the performance of acoustic models trained by combined speech database from multiple languages is better than that of models trained with speech data from a single language [Liu *et al.* 2003; Lyu *et al*. 2002]. For this reason, we use ForPA, which is an inventory of phoneme symbols, to transcribe the corpus of the two languages discussed here. Table 1 shows the statistical information of the phonemic inventory in different phonetic levels.

Table 1. The statistic information of all Mandarin (M) and Taiwanese (T) linguistic units in four levels: the numbers of Tonal Syllables (N_{TS}), Initials (N_I), Tonal Finals (N_{TF}), and context-dependent Initial/tonal Finals (N_{CDIF}). ∩and∪ mean intersection and union, respectively.

	M	T	M∪T	M∩T
N_{TS}	1288	2878	3519	647
N_I	17	19	22	14
N_{TF}	295	225	416	104
N_{CDIF}	1656	3496	4374	778

Sounds in different languages that are transcribed using the same phonemic symbols in ForPA share the same speech material. Combining two languages in this manner reduces the number of syllables by 21%. In order to easily integrate tone information, we used the context-dependent Initial and tonal Final as acoustic units, and trained these models by sharing the data which belonged to the same acoustic unit. Then, a divisive clustering algorithm was used to create context querying decision trees using four question sets, including an Initial set, a tonal Final set, the set of language properties, and a tonal information set. The above clustering approach could achieve significant improvement compared to previous results [Lyu *et al*. 2003].

Furthermore, in order to more efficiently merge the similar part of the sound for one phoneme or triphone model in both languages, we used a tying algorithm based on a decision tree to cluster the HMM models by using the maximum likelihood criterion [Liang *et al*. 1998; Lyu *et al*. 2002]. For the question sets, we used phonetic knowledge to design a total of 63 questions, including 10 language-dependent questions, 11 common questions, 28 Initial questions, and 14 Final questions. Then, the tree grew and split as we chose the optimal one among all the questions to maximize the increase in the likelihood scores or the decrease in uncertainty. Finally, the convergence condition was set to halt the growth of the decision tree. The acoustic model used in the experiment depended on the different splitting and convergence criteria adopted.

3.2 Pronunciation Modeling

The pronunciation model plays an important role in the Chinese character-based ASR engine [Liu *et al.* 2003; Huang *et al*. 2000]. It not only provides more choices during decoding if the speaker exhibits variations in pronunciation but also handles various speaking styles [Lyu *et al*. 2004]. As mentioned above, one Chinese character has more than two pronunciations in the combined phonetic inventory of Mandarin and Taiwanese. The factors of accent and regional migration can influence the pronunciation or speaking style of speakers too. Therefore, we identify the most common pronunciation variations in Taiwan in Table 2.

In Table 2, we list the five pronunciation variations that the Mandarin-Taiwanese bi-lingual recognizer can handle. Take the Chinese character "走" as an example. It is pronounced as "zau^{51}" in Taiwanese and means "to run" but is pronounced "zou^{21}" in Mandarin and means "to walk."

On the other hand, the total number of pronunciations in the pronunciation model for the decoding process is also important, because the more pronunciations are included in the lexicon, the more time the decoding process will take, and the less accurate of the ASR results will be [Strik *et al.* 1999]. The pronunciation variations will generate both improvements and deterioration in the ASR system, so previous research tried to find the optimal method to efficiently control the average pronunciation variations for one word in one language [Kesssens *et al*. 2003]. Our task is harder than that which deals with only one language. The reason is that one Chinese character must be mapped to at least two pronunciation variations, so cross-language confusion increases. In the following sections, we will propose two different methods, knowledge-based and data-driven methods, for obtaining rules of pronunciation variation.

Table 2. The five types of pronunciation variation rules in linguistic and phonological levels: 1. one orthography with pronunciation variations (OOPV); 2. colloquial literate switching (CLS) 3. tone sandhi (TS); 4. one orthography with multiple pronunciation (OOMP); 5. one pronunciation with multiple orthographies (OPMO). Other symbols and their meanings are: Chinese character (CC); Taiwanese or Mandarin pronunciations in literate style (TPL, MPL); Taiwanese or Mandarin Chinese character in colloquial style (TCC, MCC). The number [Yuen Ren Chao] following each syllable represents the tone patterns. e.g., zong^{51} means the syllable has a high-falling tone.

Within-language

(1)	CC	Base form	Surface form
OOPV	精彩	j**ing**^{55} cai^{21}	j**in**^{55} cai^{21}
	老師	lau^{21},**shii**^{55}	lau^{21} **sii**^{55}
	直的	d**it**^{55} e^{11}	d**i**^{55} e^{11}

(2)	MCC	MP	TCCL	TPL	TCCC	TPC
CLS	今天	Jin^{55}-ten^{55}	今天	gim^{33}-ten^{55}	今仔日	gin^{33}-na^{55}-lit^{55}
	明天	ming^{35}-ten^{55}	明天	bhing^{33}-ten^{55}	明仔載	mi^{33}-a^{55}-zai^{11}

(3)	CC	MP, isolated	MP, connected	TP, isolated	TP, connected
TS	總統府	zong**^{21}**,tong**^{21}** fu^{21}	zong**^{35}**-tong**^{35}**-fu^{21}	zong^{51},tong^{51} , hu^{51}	zong^{55}-tong^{55}-hu^{51}

Cross-language

(4)	CC	TP	MP
OOMP	走	zau^{51}	zou^{21}
	雨	u^{51},ho^{33}	yu^{21}
	行	giann^{15},hing^{15},hang^{15}	sing^{35},hang^{35}

(5)	Pronunciation	TCC	MCC
OPMO	jia^{55}-dan^{51}	［這裡］等	加蛋
	gau^{55}-gai^{51}	九［次］	高鈣

3.2.1 Knowledge-Based Method

As in [Wester *et al*. 2003], information about pronunciation can be derived from knowledge sources, such as pronunciation dictionaries hand-crafted by linguistic experts or extracted from the literature. In this approach, a pronunciation variation rule is simply the multiple pronunciations that appear in the lexicon for the same character. Associated probabilities can be calculated as follows. 1) the character-pronunciation pairs are derived; 2) the frequencies of the pairs are counted, and the relative frequency with respect to the total frequency of the

same Chinese character is calculated; 3) the pairs with high relative frequencies are kept as multiple pronunciation rules.

As our Mandarin knowledge source, we adopted the CKIP lexicon (http://ckip.iis.sinica.edu.tw/CKIP/) as our pronunciation lexicon source; it contains about 78,410 words. The length of one word in the lexicon varies from one Chinese character to ten, and the average of the length is 2.4 Chinese characters per word. As our Taiwanese knowledge resource, we adopted the Formosa lexicon (ForLex) [Lyu *et al*. 2000], which contains 104,179 words. The average length of one word in it is 2.8 Chinese characters. The pronunciation variation for each Chinese character was assigned a probability, which was estimated based on the frequency count of the pronunciations observed in both lexicons. The number of pronunciation variations for one Chinese character was 1.2 in the CKIP lexicon, and 2.1 in the Formosa lexicon. The number of pronunciation variations for Taiwanese was larger than that for Mandarin. The reasons are that most of the Chinese characters used in Taiwanese carry a classic literature pronunciation and a daily life pronunciation and that Taiwanese has much richer tone sandhi rules. Thus, the average number of pronunciation variations for one Chinese character is increased.

3.2.2 Data-Driven Approach

Although the regular pronunciation variations can be obtained from linguistic and phonological information, such as a dictionary, this information is not exhaustive; many phenomena in real speech have not yet been described. Therefore, another approach to deriving pronunciation variations from acoustic clues is presented below. All of the steps are also shown in Figure 5.

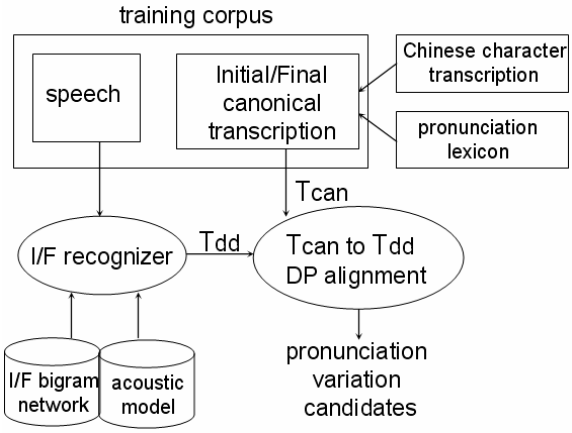

Figure 5. Diagram of pronunciation variations obtained with a data-driven approach.

First of all, the canonical transcription (T_{can}) is generated for each Chinese character in the phonetic levels of Initials and tonal Finals. Secondly, for each word in the utterance, a baseline recognition engine based on the Initial/tonal Final acoustic models is used to perform forced recognition, which adopts Viterbi search with an optional phonetic network [Strik 2003]. In this way, data-driven transcriptions (T_{dd}) of all the utterances in the training corpus can be obtained. Then, a dynamic programming algorithm is used to align T_{can} with T_{dd}. With this alignment, we can obtain a confusion table, which consists of pairs of easily confused phonetic units along with their likelihood scores.

A partial list of confusing phonetic units is shown as in Table 3. Using the above approach, we found that the major variation part in a syllable is Initial for both languages, especially in the retroflexion/un-retroflexion set. One of the possible reasons is that retroflex phonetic units exist only in Mandarin and most speakers usually do not accurately pronounce those retroflex units if their mother tongue is Taiwanese. These speakers tend to replace retroflex units with their un-retroflex counter parts.

Table 3. Some pronunciation variations obtained with the data-driven approach, where T_{can} and T_{dd} represent canonical transcription and data-driven transcription, respectively.

Mandarin				Taiwanese			
T_{can}	T_{dd}	T_{can}	T_{dd}	T_{can}	T_{dd}	T_{can}	T_{dd}
zh	z	s	sh	gh	g	p	t
sh	s	c	ch	g	d	r	l
ch	c	n	l	bh	l	h	t
z	zh	f	b	k	t	u3	u4

3.3 Searching Net

In the searching net, we use a large-vocabulary tree structured word net, because the perplexity can be reduced in the tree-structured searching net compare to the linear searching net. Figure 6 and Figure 7 show examples for a linear searching net and a tree-structured searching net, respectively. There were 5 words as searching paths in the linear net, and the equal probability of each path was set to be 1/5. We used equation 2 to calculate the entropy value based on the number of branches in each path, and we then used equation 3 to calculate the entropy from the perplexity. The perplexity of the linear searching net was found to be 5. This means that the perplexity in the linear searching net equals the number of distinct words. On the other hand, the procedure for determining the perplexity of the tree-structured searching net is described as follows. First, the Chinese characters are aligned according to their locations in multi-character words; characters that are in the same location in each word are considered to be redundant and, thus, eliminated. Finally, the entropy value is also

calculated based on the number of branches for each node, using equation (2). In the case shown in Figure 7, the entropy is 2.29, and the perplexity is 4.89, which is smaller than that of the linear searching net shown in Figure 6.

$$entropy = -\sum_i p_i \log_2 p_i \,, \tag{2}$$

$$perplexity = 2^{entropy} \,. \tag{3}$$

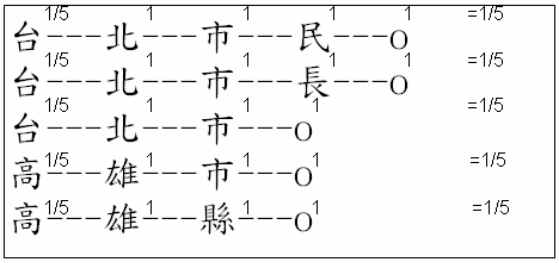

Figure 6. An example of an isolated linear searching net with its probability value.

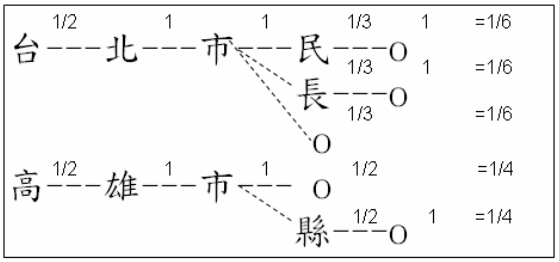

Figure 7. An example of an isolated tree-structured searching net with its probability value.

4. Experimental Results and Analysis

4.1 Corpus

All of the experiments employed a bi-lingual corpus, called ForSDa (Formosa Speech Database) [Lyu *et al.* 2004]. Both the training and testing data were read speech, which was recorded in the 16 kHz/16-bit wave-format in a normal office environment. The training set included a total of 89,164 utterances from 100 speakers, including 50 males and 50 females. Every speaker recorded speech in both languages. The utterances were phonetically balanced words, which were selected from a lexicon of about 40,000 words, using the phonetic abundant algorithm [Lyu 2003]. The length of the word varied from 1 to 6. The testing set included 2,000 utterances from 20 speakers; 10 speakers recorded speech in Taiwanese, and the other 10 speakers recorded speech in Mandarin. The statistics of the corpus employed here

are listed in Table 4.

Table 4. Statistics of the bi-lingual speech corpus used for training and testing sets. M: Mandarin, T: Taiwanese.

	Langue ID.	No. of Speakers	No. of Words	No. of Hours
Training	M	100	43078	11.3
	T	100	46086	11.2
Test_M	M	10	1000	0.28
Test_T	T	10	1000	0.28

4.2 Experimental Setup

The experiment setup can be described as follows. Firstly, we used context dependent Initials and tonal Finals with 16 Gaussian mixtures in HMM modeling. The feature vectors used in the HMM included 42 components, with 12 mel-frequency cepstral coefficients (MFCCs), normalized log energy, and pitch with their first and second order derivatives. Secondly, in pronunciation modeling, we used three models, which included knowledge-based, data-driven, and combined approaches, called P_{KW}, P_{DD} and P_{KW+DD}, respectively. The average number of pronunciations for one Chinese character for each pronunciation lexicon was 3.2, 2.7 and 3.9 for P_{KW}, P_{DD} and P_{KW+DD}, respectively. Finally, the tree-structured searching net consisted of 30,000 words, and the word perplexity of the net was 15,249. This means that there where almost 15,249 candidates for each input speech utterance in the decoding phase. Additionally, the output of the recognizer was Chinese characters; therefore, we evaluated the performance based on the Chinese character error rate (CER).

4.3 Experiment Results

Table 5 shows the CER results for pronunciation modeling with the Taiwanese and Mandarin testing sets. We can draw two conclusions; firstly, when the pronunciation model P_{KW+DD} was used, the CER was minimal for both languages. The reason is that P_{KW+DD} could capture both within-language and cross-language pronunciation variations. Secondly, the CER of the Test_M set with P_{DD} was better than that with P_{KW}, but the CER of the Test_T set was worse. A possible reason is that most of the pronunciation variations in Taiwanese can be found in the dictionary or lexicon source, such as tone sandhi or colloquial/literate switching. However, in Mandarin, most of the pronunciation variations are due to co-articulation, regional accents, speaking rates, speaking styles, etc. Such types of the variation can only be captured in speech data, not in lexicons. Therefore, the CER of Test_M dropped about 2.2% (17.9%-20.1%) when P_{DD} was used compared to the result obtained with P_{KW}, but the CER of Test_T increased 0.7% (18.3%-17.6%).

Table 5. CER (Character Error Rate) results for three pronunciation models with two testing sets. P_{KW}: pronunciation modeling using the knowledge-based method; P_{DD}: pronunciation modeling using the data-driven approach; P_{KW+DD}: pronunciation modeling using both P_{KW} and P_{DD}.

	P_{KW}	P_{DD}	P_{KW+DD}
Test_M	20.1%	17.9%	16.2%
Test_T	17.6%	18.3%	15.0%

4.4 Error Analysis

The addition of pronunciation variants to a lexicon increases the confusability, especially if the lexicon is large. Here, the large increase in confusability was probably the reason why only a small improvement or even deterioration in performance is found. The experimental results represented in Figure 8 show the CER performance as a function of the number of pronunciation variations for each Chinese character. It can be seen that the CER decreased when the average number of pronunciation variations increased. The lowest CER results were obtained when the number of pronunciation averaged 3.9. This was achieved using P_{KW+DD} and by eliminating variants with probabilities smaller than 0.1.

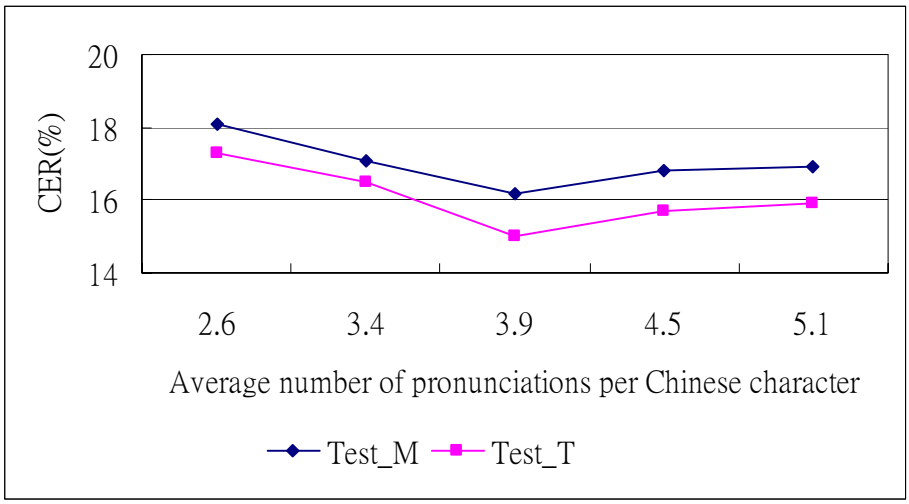

Figure 8. CER performance for P_{KW+DD} with different numbers of pronunciation variations per Chinese character.

Moreover, the error types mentioned above can be classified into the following 3 sets.

A. Cross-language homophonic confusion

This kind of error is just like the fifth term in Table 1, and occurs when different Chinese words belonging to different languages have the same or similar pronunciation. Therefore, the

confusion of choosing the final Chinese words will occur during the decoding phase. For example, the pronunciation of the Chinese word "星系" in Mandarin, that is, /sing^{55}-si^{51}/, is similar with that of "先死" in Taiwanese, that is, /sing^{33}-si^{51}/. The same is true of "高等" in Mandarin, pronounced /gau^{55}-dng^{13}/, and "教堂" in Taiwanese, pronounced, /gau^{51}-dng^{13}/.

B. Within-language homophonic confusion

This type of error is similar to the first error type, but it only occurs within one language. For example, the Chinese words "穢亂" and "會亂" have the same pronunciation, that is, /huei^{51}-luan^{51}/, in Mandarin, and "交待" and "交代" both have the same pronunciation in Mandarin, that is, /jiau^{55}-dai^{51}/, and in Taiwanese, that is, /gau^{55}-dai^{55}/.

C. Tone confusion:

This kind of error occurs due to mismatch between the tone pattern and speech features. We add the tone vectors to the feature parameters, the words, "水餃" and "睡覺" can be easily discriminated a tonal phase. However, there is also a side effect if the acoustic model in the tone aspect is not robust enough. A major tone error may be due to confusion between a high-level (55) tone and a mid-level (35) tone. Another major error may due to the confusion between a mid-falling (31) tone and a high-falling tone. Following are some tone confusion examples:

1) "縫補" /fng^{35}-bu^{31}/ and "蜂舞" /fng^{55}-u^{31}/.
2) "股票" /gu^{31}-piau^{51}/ and "顧票" /gu^{51}-piau^{51}/.

Most of the performance deterioration observed in this experiment was caused by the above error types; however, the performances of deterioration are smaller than that of improvements by adding pronunciation variations to the lexicon. Therefore, finally, we got an improvement in CER result.

5. Conclusion

As mentioned in the introduction, the goal of this study was to convert both Taiwanese and Mandarin speech into Chinese characters. In order to deal with the issues of multiple pronunciations and pronunciation variations for each Chinese character in these two languages in the ASR system, we developed a one-pass, three-layer recognizer, which includes combined bi-lingual acoustic models, an integrated pronunciation model and a tree-structure-based searching net. In the pronunciation model, an integrated method is used to combine the knowledge-based and data-driven approaches. Since the knowledge-based approach is used, homophony in Chinese characters can be addressed, and since the data-driven approach is employed, speakers' accents or styles can also be dealt with.

The experimental results showed that the CER could be improved by using the three different pronunciation models. The best performance was 16.2% and 15.0% for the testing sets Test_M and Test_T, respectively, where the perplexity was 15,249 for 30,000 words, and the P_{KW+DD} pronunciation model was used. In addition, in order to limit the side effect where in the increase in the size of the pronunciation lexicon causes the performance to deteriorate, the average number of pronunciations for both languages was 3.9.

The method proposed in this paper has been applied to two languages in the Chinese language family, but it can be easily extended to other languages or dialects. We have also discussed the major five pronunciation variations found in Taiwan. This is the first work, to the best of our knowledge, that has systemically investigated pronunciation variations in Mandarin and Taiwanese speech conversion to Chinese characters using ASR technology.

References

Aubert, X., "One pass cross word decoding for large vocabularies based on a lexical tree search organization," In *Proceedings of the European Conference on Speech Communication and Technology*, 1999, Budapest, Hungary, pp. 1559-1562.

Bacchiani, M., and M. Ostendorf, "Joint lexicon, acoustic unit inventory and model design," *International Journal of Speech Communication*, 29(2-4), 1999, pp. 99-114.

Chao, Y. R., Tone contour, http://en.wikipedia.org/wiki/Tone_contour/, 1979.

Cremelie, N., and J.-P. Martens, "In search of pronunciation rules," In *Proceedings of the European Speech Communication Association (ESCA) Workshop on Modeling Pronunciation Variation for Acoustic Speech Recognition*, 1998, Rolduc, Kerkrade, pp. 103-108.

Downey, S., and R. Wiseman, "Dynamic and static improvements to lexical baseforms," In *Proceedings of the Workshop on Modeling Pronunciation Variations*, 1998, Roldue, pp. 157-162.

Finke, M., and A. Waibel, "Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition," In *Proceedings of the European Conference on Speech Communication and Technology*, 1997, Rhodos, Greece, pp. 2379-2382.

Fukada, T., and Y. Sagisaka, "Automatic generation of a pronunciation dictionary based on a pronunciation network," In *Proceedings of the European Conference on Speech Communication and Technology*, 1997, Rhodos, pp. 2471-2474.

Fukada, T., T. Yoshimura, and Y. Sagisaka, "Automatic generation of multiple pronunciations based on neural networks and language statistics," In *Proceedings of the European Speech Communication Association (ESCA) Workshop on Modeling Pronunciation Variation for Acoustic Speech Recognition*, 1998, Rolduc, Kerkrade, pp. 103-108.

Holter, T., and T. Svendsen, "Maximum likelihood modelling of pronunciation variation," *International Journal of Speech Communication*, 29, 1999, pp. 177-191.

Huang, C., E. Chang, J.L. Zhou, and K.F. Lee, "Accent Modeling Based on Pronunciation Dictionary Adaptation for Large Vocabulary Mandarin Speech recognition," In *Proceedings of the International Conference on Spoken Language Processing*, 2000, Beijing.

Jurafsky, D., W. Ward, J. Zhang, K. Herold, X. Yu, and S. Zhang, "What kind of pronunciation variation is hard for triphones to model?" In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2001, Salt Lake City, Utah, pp. 577-580.

Kam, P., and T. Lee, "Modeling pronunciation variation for Cantonese speech recognition," In *Proceedings of ISCA ITR-Workshop on Pronunciation Modeling and Lexicon Adaptation*, 2002, Colorado, USA, pp.12-17.

Kam, P., T. Lee, and F. Soong, "Modeling Cantonese pronunciation variation by acoustic model refinement," In *Proceedings of the 8th European Conference on Speech Communication and Technology*, 2003, Geneva, Switzerland, pp.1477-1480.

Kessens, J.M., C. Cucchiarini, and H. Strik, "A data-driven method for modeling pronunciation variation," *International Journal of Speech Communication*, 40, 2003, pp. 517-534.

Kessens, J.M., H. Strik, and C. Cucchiarini, "Modeling pronunciation variation for ASR: Comparing criteria for rule selection," In *Proceedings of the Workshop on Pronunciation Modeling and Lexicon Adaptation*, 2002, Estes Park, USA, pp. 18-23.

Kessens, J.M., M. Wester, and H. Strik, "Improving the Performance of a Dutch CSR by Modeling Within-word and Cross-word Pronunciation Variation," *International Journal of Speech Communication on Special issue of 'Modeling Pronunciation Variation for Automatic Speech Recognition'*, 29(2-4), 1999, pp. 193-207.

Kipp, A., M.-B. Wesenick, and F. Schiel, "Automatic detection and segmentation of pronunciation variants in German speech corpora," In *Proceedings of the International Conference on Spoken Language Processing*, 1996, Philadelphia, USA, pp. 106-109.

Lee, T., W. Lau, Y. W. Wong, and P.C. Ching, "Using tone Information In Cantonese Continuous Speech Recognition," *ACM Transactions on Asian Language Information Processing*, 1, 2002, pp. 83-102.

Liang, M.S., R.Y. Lyu, and Y.C. Chiang, "An efficient algorithm to select phonetically balanced scripts for constructing corpus," In *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering,* 2003, Beijing, China.

Liang, P.Y. , J. L. Shen, and L. S. Lee, "Decision Tree Clustering for Acoustic Modeling in Speaker-Independent Mandarin Telephone Speech Recognition," In *Proceedings of the International Symposium on Chinese Spoken Language Processing* , 1998, Singapore, pp. 207-211.

Liao, Y. F., N. Wang, M. Huang, H. Huang, and F. Seide, "Improvements of the Philips 2000 Taiwan Mandarin Benchmark System," In *Proceedings of the International Conference on Spoken Language Processing*, 2000, Beijing. pp. 298-301.

Liu, Y., and P. Fung, "Modeling partial pronuncia-tion variations for spontaneous Mandarin speech recog-nition," *International Journal of Computer Speech and Language*, 17, 2003, pp. 357-379.

Liu, Y., and P. Fung, "Partial change accent models for accented Mandarin speech recognition," In *Proceedings of the IEEE Workshop on ASRU*, 2003, St. Thomas, U.S. Virgin Islands.

Liu, Y., and P. Fung, "State-Dependent Phonetic Tied Mixtures with Pronunciation Modeling for Spontaneous Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, 12, 2004, pp. 351-364.

Lyu, D.C., B.H. Yang, M.S. Liang, R.Y. Lyu, and C.N. Hsu, "Speaker Independent Acoustic Modeling for Large Vocabulary Bi-lingual Taiwanese/Mandarin Continuous Speech Recognition," *In Proceedings of the 9th Australian International Conference on Speech Science & Technology*, 2002, Melbourne, Australia.

Lyu, D.C., M.S. Liang, Y.C. Chiang, C.N. Hsu, and R.Y. Lyu, "Large Vocabulary Taiwanese (Min-nan) Speech Recognition Using Tone Features and Statistical Pronunciation Modeling," In *Proceedings of the 8th European Conference on Speech Communication and Technology*, 2003, Geneva, Switzerland.

Lyu, D.C., M.S. Liang, Y.C. Chiang, C.N. Hsu and R.Y. Lyu, "Large Vocabulary Taiwanese (Min-nan) Speech Recognition Using Tone Features and Statistical Pronunciation Modeling," In *Proceedings of the European Conference on Speech Communication and Technology*, 2003, Geneva, Switzerland.

Lyu, R.Y., C.Y. Chen, Y.C. Chiang, and M.S. Liang, "Bi-lingual Manda-rin/Taiwanese(Min-nan), Large Vocabulary, Continuous Speech Recognition System Based on the Yong-yong Phonetic Alphabet," *In Proceedings of the International Conference on Spoken Language Processing*, 2000, Beijing, China.

Lyu, R.Y., D.C. Lyu, M.S. Liang, M.H. Wang, Y.C. Chiang, and C.N. Hsu, "A Unified Framework for Large Vocabulary Speech Recognition of Mutually Unintelligible Chinese "Regionalects"," In *Proceedings of the 8th International Conference on Spoken Language Processing*, 2004, Jeju Island, Korea.

Lyu, R.Y., M.S. Liang, and Y.C. Chiang, "Toward Constructing A Multilingual Speech Corpus for Taiwanese (Minnan), Hakka, and Mandarin," *International Journal of Computational Linguistics and Chinese Language Processing*, 9(2), 2004, pp. 1-12.

Odell, J.J., V. Valtchev, P.C.Woodland, and S.J.Young, "A One Pass Decoder Design for Large Vocabulary Recognition," In *Proceedings of Human Language Technology Workshop*, 1994, pp. 405-410.

Peters, S.D., and P. Stubley, "Visualizing speech trajectories," In *Proceedings of the European Speech Communication Association (ESCA) Workshop on Modeling Pronunciation Variation for Acoustic Speech Recognition*, 1998, Rolduc, Kerkrade, pp. 103-108.

Polzin, T.S., and A.H. Waibel, "Pronunciation variations in emotional speech," In *Proceedings of the European Speech Communication Association (ESCA) Workshop on*

Modeling Pronunciation Variation for Acoustic Speech Recognition, 1998, Rolduc, Kerkrade, pp. 103-108.

Riley, M., W. Byrne, M. Finke, S. Khudanpur, A. Ljolje, J. McDonough, H. Nock, M. Saraclar, C. Wooters, and G. Zavaliagkos, "Stochastic pronunciation modelling from hand-labelled phonetic corpora," *International Journal of Speech Communication*, 29, 1999, pp. 209-224.

Singh, R., B. Raj, and R. Stern, "Automatic generation of subword units for speech recognition systems," *IEEE Transactions on Speech and Audio Processing*, 10, 2002, pp. 89-99.

Soltau, H., F. Metze, C. Fuegen, and A. Waibel, "A One-pass decoder based on polymorphic linguistic context assignment," In *Proceedings of Automctic Speech Recognition and Understanding Workshop*, 2001, Trento, Italy.

Strik, H., and C. Cucchiarini, "Modeling Pronunciation Variation for ASR: Overview and Comparison of Method," *International Journal of Speech Communication*, 29, 1999, pp. 225-246.

Strik H., J.M. Kessens, and M. Wester, "Modeling Pronunciation Variation for Automatic Speech Recognition,", In *Proceedings of the European Speech Communication Association (ESCA) workshop*, 1998, Rolduc, Kerkrade, pp. 137-144.

Torre, D., L. Villarrubia, L. Hernandez, and J.M. Elvira, "Automatic Alternative Transcription Generation and Vocabulary Selection for Flexible Word Recognizers," In *Proceedings of the International Conference on Acoustic*s, *Speech, and Signal Processing*, 1997, Munich, pp. 1463-1466.

Wester, M., and E. Fosler-Lussier, "A comparison of data-derived and knowlegde-based modeling of pronunciation variation," In *Proceedings of International Conference on Spoken Language Processing*, 2000, Beijing, China, pp. 270-273.

Wester, M., "Pronunciation Modeling for ASR knowledge-based, Data-driven Methods," *International Journal of Computer Speech and Language*, 88, 2003, pp. 69-85.

Wester, M., J.M. Kessens, and H. Strik, "Pronunciation Variation in ASR: Which Variation to model?" In *Proceedings of the International Conference on Spoken Language Processing*, 2000, Beijing, China, pp. 488-491.

Yang, Q., and J.-P. Martens, "Data driven lexical modeling of pronunciation variation in ASR," In *Proceedings of the International Conference on Spoken Language Processing*, 2000, Beijing, China, pp. 417-420.

Zeppenfeld, T., M. Finke, K. Ries, M. Westphal, and A. Waibel, "Recognition of conversational speech using the JANUS speech engine," In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1997, Munich, pp. 1815-1818.

Computational Linguistics and Chinese Language Processing
Vol. 10, No. 3, September 2005, pp. 381-396

Chinese Word Segmentation by Classification of Characters

Chooi-Ling Goh*, Masayuki Asahara* and Yuji Matsumoto*

Abstract

During the process of Chinese word segmentation, two main problems occur: segmentation ambiguities and unknown word occurrences. This paper describes a method to solve the segmentation problem. First, we use a dictionary-based approach to segment the text. We apply the Maximum Matching algorithm to segment the text forwards (FMM) and backwards (BMM). Based on the difference between FMM and BMM, and the context, we apply a classification method based on Support Vector Machines to re-assign the word boundaries. In so doing, we use the output of a dictionary-based approach, and then apply a machine-learning-based approach to solve the segmentation problem. Experimental results show that our model can achieve an F-measure of 99.0 for overall segmentation, given the condition that there are no unknown words in the text, and an F-measure of 95.1 if unknown words exist.

Keywords: Chinese, word segmentation, segmentation ambiguity, unknown word, maximum matching algorithm, support vector machines

1. Introduction

The first step in Chinese information processing is word segmentation. This is because in written Chinese, all characters are joined together, and there are no separators to mark word boundaries. A similar problem also occurs with languages like Japanese, but at least with Japanese, there are three types of characters (hiragana, katakana and kanji). This helps provide clues for finding word boundaries. In the case of Chinese, as there is only one type of character (hanzi), more segmentation ambiguities may occur in a text. During the process of segmentation, two main problems are encountered: segmentation ambiguities and unknown word occurrences. This paper focuses on solving the segmentation ambiguity problem and proposes a sub-model to solve the unknown word problem. There are basically two types of segmentation ambiguity: covering ambiguity and overlapping ambiguity. The definitions are

* Graduate School of Information Science, Nara Institute of Science and Technology, Japan
E-mail: {ling-g, masayu-a, matsu}@is.naist.jp

[Received August 31, 2004; Revised February 21, 2005; Accepted February 21, 2005]

given below.

Let x, y, z be some strings which could consist of one or more Chinese characters. Assuming that W is a given dictionary, the covering ambiguity is defined as follows: For a string $w = xy$, $x \in W$, $y \in W$, and $w \in W$. As almost any single character in Chinese can be considered as a word, the above definition reflects only those cases where both word boundaries .../*xy*/... and .../*x*/*y*/... can be found in sentences. On the other hand, overlapping ambiguity is defined as follows: For a string $w = xyz$, both $w_1 = xy \in W$ and $w_2 = yz \in W$ hold. Although most of the time, one form of segmentation is preferred over the other, we still need to know about the contexts in which the other form is used. Both types of ambiguity require that the context be considered to decide which is the correct segmentation form given a particular occurrence in the text.

(1a) and (1b) show examples of covering ambiguity. The string "一家" is treated as a word in (1a) but as two words in (1b).

(1a)胡/世庆/一家/三/口/
Hu/ Shiqing/ whole family/ three/ member
(All three members of Hu Shiqing's family)

(1b)在/巴黎/一/家/杂志/上/
in/ Paris/ one/ company/ magazine/ at/
(At one magazine company in Paris)

On the other hand, (2a) and (2b) are examples of overlapping ambiguity. The string "不可以" is segmented as "不/可以" in (2a) and as "不可/以" in (2b), according to the context in each sentence.

(2a)不/可以/淡忘/远在/故乡/的/父母/
not/ can/ forget/ far away/ hometown/ DE/ parents/
(Cannot forget parents who are far away at home)

(2b)不可/以/营利/为/目的/
cannot/ by/ profit/ be/ intention
(Cannot have the intention to make a profit)

We intend to solve the ambiguity problems by combining a dictionary-based approach with a statistical model. In so doing, we make use of the information in a dictionary in a statistical approach. The Maximum Matching (MM) algorithm, a very early and simple dictionary-based approach, is used to initially segment the text by referring to a dictionary. It tries to match the longest possible words found in the dictionary. We can parse a sentence either forwards or backwards. Normally, the differences between the results of forward and backward parsing will indicate the locations where overlapping ambiguities occur. Then, we use a Support Vector Machine-based (SVM) classifier to decide which output should be the correct answer. As for covering ambiguities, in most cases, forward and backward MM will give the same output. In this case, we just make use of the contexts to decide whether or not to split a word into two or more words. Our experimental results show that the proposed method can solve 92% of overlapping ambiguities and 52% of covering ambiguities.

2. Previous Works

Solving the ambiguity problems is a fundamental task in Chinese segmentation process. Although many previous researches have focused on segmentation, only a few have reported on the accuracy achieved in solving ambiguity problems. Li *et al*. [2003] proposed an unsupervised method for training Naïve Bayes classifiers to resolve overlapping ambiguities. They achieved 94.13% accuracy in 5,759 cases of ambiguity. An alternative form of TF.IDF weighting was proposed for solving the covering ambiguity problem in [Luo *et al.* 2002]. They focused on 90 ambiguous words and achieved an accuracy of 96.58%.

Most of the previous methods reported on the accuracy of overall segmentation. Recently, many researches have adopted multiple models. Furthermore, most researchers have realized that character-based approaches are more effective than word-based approaches to Chinese word segmentation. In [Xue and Converse 2002], two classifiers were combined to perform Chinese word segmentation. First, a Maximum Entropy model was used to segment the text, and then an error driven transformation model was used to correct the word boundaries. Their method also used character-based tagging to assign the positions of characters in words. They achieved an F-measure of 95.17 using the Penn Chinese Treebank. Another recent study was that of Fu and Luke [2003], who proposed hybrid models for integrated segmentation. Modified word juncture models and word-formation patterns were used to find word boundaries and at the same time to identify unknown words. They achieved and F-measure of 96.1 using the Peking University Corpus. As the above studies used different corpora in their experiments, it is difficult to tell which method performed better.

Solving the unknown word problem is also an important step in word segmentation. An unknown word is a word not found in a dictionary. Therefore, it cannot be segmented correctly by simply referring to the dictionary. Many approaches for unknown word detection

have been proposed [Chen and Bai 1997; Chen and Ma 2002; Fu and Wang 1999; Lai and Wu 1999; Ma and Chen 2003; Nie *et al*. 1995; Shen *et al.* 1998; Zhang *et al*. 2002; Zhou and Lua 1997]. These include rule-based, statistics-based, and hybrid models. We cannot ignore the unknown word problem since there are always some unknown words (such as person names, numbers etc.) in a text even when we use a very large dictionary. The creation of new words in Chinese is a continuous process. For example, names for new diseases, technical terms, and new expressions are always being created. The accuracy is better if one focuses only on certain types of unknown words such as person names, place names, or transliteration names, when accuracy of over 80% can be achieved. However, for general unknown words, such as common nouns, verbs etc., the accuracy ranges from only 50% to 70%.

3. Proposed Method

We propose a method that uses only minimum resources, meaning that only a segmented corpus is required. The underlying concept of our proposed method is as follows. We regard the problem as a character classification problem. We believe that each character in Chinese tends to appear in certain positions in words. A character can be used at the beginning of a word, in the middle of a word, at the end of a word, or as a single-character word. It can appear at different positions in different words. By looking at the usage of the characters, we can decide on their position tags using a machine learning based model, which in our case is the Support Vector Machines model [Vapnik 1995]. Our method employs a model to solve the ambiguity problem and, at the same time, embeds a model to detect unknown words. We will next describe the method in more detail in the following section.

3.1 Maximum Matching Algorithm

We intend to solve the ambiguity problem by combining a dictionary-based approach with a statistical model. The Maximum Matching (MM) algorithm is regarded as the simplest dictionary-based word segmentation approach. It starts from one end of a sentence and tries to match the first longest word wherever possible. It is a greedy algorithm, but it has been empirically proved to achieve over 90% accuracy if the dictionary used is large. However, the ambiguity problem cannot be solved effectively, and it is impossible to detect unknown words because only those words existing in the dictionary can be segmented correctly. If we look at the outputs produced by segmenting the sentence forwards (FMM), from the beginning of the sentence, and backwards (BMM), from the end of the sentence, we can determine the places where overlapping ambiguities occur. For example, FMM will segment the string “即将来临时” (when the time comes) into “即将/来临/时/”(immediately/ come/ when), but BMM will segment it into “即/将来/临时/”(that/ future/ temporary).

Let O_f and O_b be the outputs of FMM and BMM, respectively. According to Huang

[1997], for overlapping cases, if $O_f = O_b$, then the probability that both the MMs will be the correct answer is 99%. If $O_f \neq O_b$, then the probability that either O_f or O_b will be the correct answer is also 99%. However, for covering ambiguity cases, even if $O_f = O_b$, both O_f and O_b could be correct or could be wrong. If there exist unknown words, they normally will be segmented as single characters by both FMM and BMM. Based on the differences and contexts created by FMM and BMM, we apply a machine learning based model to re-assign the position tags which indicate character positions in words.

3.2 Support Vector Machines

Support Vector Machines (SVM) [Vapnik 1995] are binary classifiers that search for a hyperplane with the largest possible margin between positive and negative samples (see Figure 1). Suppose we have a set of training data for a binary class problem: $(\mathbf{x}_1, y_1),\ldots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in R^n$ is the feature vector of the ith sample in the training data and $y_i \in \{+1, -1\}$ is its label. The goal is to find a decision function which accurately predicts the label y for an unseen $\mathbf{x}$. An SVM classifier gives a decision function $f(\mathbf{x})$ for an input vector $\mathbf{x}$, where

$$f(\mathbf{x}) = sign\left(\sum_{\mathbf{z}_i \in SV} \alpha_i y_i K(\mathbf{x}, \mathbf{z}_i) + b\right).$$

$f(\mathbf{x})$= +1 means that $\mathbf{x}$ is a positive member, and $f(\mathbf{x})$ = -1 means that $\mathbf{x}$ is a negative member. The vectors $\mathbf{z}_i$ are called support vectors, and they are assigned a non-zero weight α_i. Support vectors and the parameters are determined by solving a quadratic programming problem. $K(\mathbf{x}, \mathbf{z})$ is a kernel function which computes an extended inner product of input vectors. We use a polynomial kernel function of degree 2, that is, $K(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x}\cdot \mathbf{z})^2$.

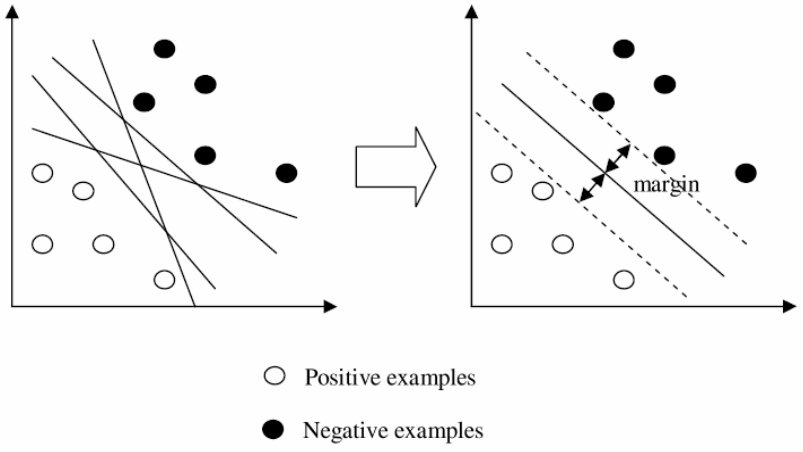

Figure 1. Maximizing the margin

We use *YamCha* [Kudo and Matsumoto 2001] to train our SVM models. *YamCha* is an SVM-based multi-purpose chunker. It extends binary classification to *n*-class classification for natural language processing purposes, where we would normally want to classify the words into several classes, as in the case of POS tagging or base phrase chunking. Two straightforward methods are mainly used for this extension, the "one-vs-rest" method and the "pairwise" method. In the "one-vs-rest" method, *n* binary classifiers are used to compare one class with the rest of the classes. In the "pairwise" method, $\binom{n}{2}$ binary classifiers are used to compare between all pairs of classes. We need to classify the characters into 4 categories (B, I, E or S, as shown in Table 1) in our method. We used the "pairwise" classification method in our experiments because it is more efficient during the training phase. Details of the system can be found in [Kudo and Matsumoto 2001].

Table 1. Position tags in a word (BIES tags)

Tag	Description
S	one-character word
B	first character in a multi-character word
I	intermediate character in a multi-character word (for words longer than two characters)
E	last character in a multi-character word

3.3 Classification of Characters

We intend to classify the characters using the SVM-based chunker [Kudo and Matsumoto 2001] as described in Section 3.2. [Xue and Converse 2002] proposed to regard the word segmentation problem as a character tagging problem. Instead of segmenting a sentence into word sequences directly, characters are first assigned with position tags. Later, based on these postion tags, the characters are converted into word sequences. The basic features used are the characters. However, the number of examples per feature will be small if there is only character information and no other information is provided. Since there are always more known words than unknown words in a text, it is advantageous if we can segment known words beforehand. Therefore, we supply the outputs from FMM and BMM as some of the features. In this case, the learning by SVM is guided by a dictionary for known word segmentation. The similarities and differences between FMM and BMM are used to train the SVM to solve the segmentation ambiguity problem.

First, we convert the output of the MMs into a character-wise form, where each character is assigned a position tag as described in Table 1. The BIES tags are as described in [Uchimoto *et al*. 2000] and [Sang and Veenstra 1999] for named entity extraction. These tags show possible character positions in words. For example, the character "本" is used as a single character word in "一/本/书/" (a book), at the end of a word in "剧本' (script), at the

beginning of a word in “本来” (originally), or in the middle of a word in “基本上” (basically).

The solid box in Figure 2 shows the features used to determine the tag of the character “春” at location *i*. In other words, our feature set consists of the characters, the FMM and BMM outputs, and the previously tagged outputs. The context window is two characters on both the left and right sides of the current character. Based on the output position tags, finally, we get the segmentation “迎/新春/联谊会/上/” (welcome/ new year/ get-together party/ at/).

Position	Char.	FMM	BMM	Output
i-2	迎	B	S	S
i-1	新	E	B	B
i	春	B	E	E
i+1	联	E	B	B
i+2	谊	S	E	I
i+3	会	B	B	E
i+4	上	E	E	S

Figure 2. An illustration of classification process applied to “At the New Year gathering party”

4. Experiments and Results

We run our experiments with two datasets, the PKU Corpus and the SIGHAN Bakeoff data. The evaluation was conducted using the tool provided in SIGHAN Bakeoff [Sproat and Emerson 2003].

4.1 Experiment with the PKU Corpus

4.1.1 Accuracy on Solving Ambiguity Problem

The corpus used for this experiment was provided by Peking University (PKU)[1] and consists of about 1.1 million words. It is a segmented and POS-tagged corpus, but we only used the segmentation information for our experiments. We divided the corpus randomly into two parts consisting of 80% and 20% of the corpus, for training and testing, respectively. Since our purpose in this experiment was only to solve the ambiguity problem, not the unknown word

[1] Institute of Computational Linguistics, Peking University, http://www.icl.pku.edu.cn/

detection problem, we assumed that all the words could be found in the dictionary. We created a dictionary with all the words from the corpus, which had 62,030 entries (referred to as Experiment 1). This experiment was conducted to evaluate the performance of the method in solving the ambiguity problem.

It is difficult to determine how many ambiguities appear in a sentence. For example, in the sentence shown in Figure 2, "迎新" (welcome the new year), "新春" (new year), "春联" (a strip of red paper that is pasted beside a door; on it is written some greeting words to celebrate the new year in China), "联谊" (get-together), "联谊会" (get-together party), "会上" (at the meeting) and "上" (at) are all possible words. A word candidate may cause more than one ambiguity with the alternative word candidates. Therefore, we try to represent the ambiguities by means of character units since our method is character-based. We assign each character to one of these six categories. Let,

O_f = Output of FMM,
O_b = Output of BMM,
Ans = Correct answer,
Out = Output from our system.

Table 2. Disambiguation results obtained with the PKU Corpus

Category	Conditions	No. of Char.	Percentage
Allcorrect	$O_f = O_b = Ans = Out$	330220	96.35%
Correct	$O_f \neq O_b$ and $Ans = Out$	7663	2.23%
Wrong	$O_f \neq O_b$ and $Ans \neq Out$	658	0.19%
Match	$O_f = O_b$ and $O_f \neq Ans$ and $Ans = Out$	1876	0.55%
Mismatch	$O_f = O_b$ and $O_f \neq Ans$ and $Ans \neq Out$	1738	0.51%
Allwrong	$O_f = O_b = Ans$ and $Ans \neq Out$	571	0.17%
Total		342726	100.00%

Table 2 shows the conditions for each category together with the results obtained with the method for solving the ambiguity problem. The categories *Allcorrect, Correct*, and *Match* have correct answers, whereas the categories *Wrong, Mismatch*, and *Allwrong* have wrong answers. We can roughly say that the categories *Correct* and *Wrong* contain overlapping ambiguities, and that the categories *Match, Mismatch*, and *Allwrong* contain covering ambiguities. We can also say that *Match* and *Mismatch* categories refer to cases where words should be split, whereas *Allwrong* category refers to cases where words should not be split but the system mistakenly splits them.

Overall, we could correctly tag 99.13% of the characters. If we only consider the overlapping cases (*Correct* and *Wrong*), 92.09% of the characters were correctly tagged. As for covering cases, if we look at only those cases where we need to split the words (*Match* and *Mismatch*), then 51.91% of them were successfully split.

Table 3. Segmentation results obtained with the PKU Corpus

	FMM	BMM	SVM (char. only)	FMM +SVM	BMM +SVM	FMM+BMM+SVM (=Experiment 1)
Recall	96.9	97.1	94.0	98.7	98.7	**98.9**
Precision	97.7	97.9	94.3	98.9	99.0	**99.1**
F-measure	97.3	97.5	94.1	98.8	98.9	**99.0**

Table 3 shows overall word segmentation results. Compared with the baseline models, namely, FMM, BMM, and SVM (using only characters as features), our proposed method can achieve higher accuracy with an F-measure of 99.0. This means that our method is able to solve the ambiguity problem given information about locations where ambiguities occur by looking at the outputs of FMM and BMM.

4.1.2 Accuracy in Solving the Unknown Word Problem

The corpus used in this experiment was the same as that described in Section 4.1.1, but the setting is different. In this round, we divided the corpus into three sets, referred to as Set 1, Set 2, and Set 3. Set 1 plus Set 2 (80%) was used for training, and Set 3 (20%) was used for testing, just as in the previous experiment. The difference was in the preparation of the dictionary. It was prepared in two ways. In the first case, all the words from Set 1 and Set 2 were used to create the dictionary. There were 49,433 entries in the dictionary and 8,346 (4.0%) unknown words in the testing data (referred to as Experiment 2). This experiment was conducted to investigate the performance of the method when unknown words exist. In the second case, only the words from Set 1 were used to create the dictionary, resulting in a situation where unknown words existed in the training data (referred to as Experiment 3). The top part of Table 4 shows the proportions of Set 1 and Set 2, along with the sizes of the dictionaries and the numbers of unknown words in Set 2 and Set 3 (the testing data). Set 2 served as a learning model for unknown word detection[2]. When we segmented Set 2 using FMM and BMM, most of the unknown words were segmented into single characters (namely tag 'S'). Based on these tags and contexts, the SVM-based chunker was trained to change the

[2] It is possible to create unknown word phenomena in a training corpus by collecting all the words from the corpus but dropping some words like compounds, proper names, numbers etc. However, since we assume that out target corpus is only a segmented corpus, without other information like POS tags, it is difficult to determine what words that should be dropped and be treated as unknown words.

tags into the correct answers. The last experiment (referred to as Experiment 4) was the opposite of Experiment 2; nothing was used to create the dictionary. All the words were considered to be unknown words. Only the characters were used as features during the classification phase, meaning that no information from FMM and BMM was available.

Table 4. Different settings and segmentation results with unknown words (PKU Corpus)

	Experiment 1	Experiment 2	Experiment 3			Experiment 4
Set 1(%)/ Set 2(%)		80/0	60/20	40/40	20/60	0/80
# of words in Dict.	62,030	49,433	41,582	33,355	22,363	0
# of unk-words in Set 2	0	0	10,927	25,297	53,353	All
# of unk-words in Test(Set 3)	0	8,346	9,768	11,924	17,115	All
Recall	98.9	95.3	**95.8**	95.7	95.2	94.0
Precision	99.1	90.7	93.5	94.5	**94.7**	94.3
F-measure	99.0	92.9	94.7	**95.1**	94.9	94.1
OOV(recall)	-	8.0	41.2	54.9	63.3	**69.3**
IV(recall)	98.9	**98.9**	98.1	97.4	96.5	95.0

The bottom part of Table 4 shows the results obtained in these experiments. Our method in fact worked quite well in solving both the segmentation ambiguity and unknown word detection problems. However, while the accuracy for unknown word detection improved, the performance in solving the ambiguity problem worsened. This is because the precision in unknown word detection was not one hundred percent. False unknown words caused the accuracy of known word segmentation to deteriorate. The highest recall rate that we could get for known words was 98.9% (as in model 80/0) and that for unknown words was 69.3% (as in model 80/0). However, the best overall segmentation result was achieved by dividing the training corpus in half (as in model 40/40), and the result was an F-measure of 95.1. This is the optimal point where a balance is found between detecting unknown words and at the same time maintaining accuracy in the segmentation of known words. Figure 3 shows the F-measure results for segmentation and recall results for unknown words and known words, when different proportions of the training corpus were used to create the dictionary.

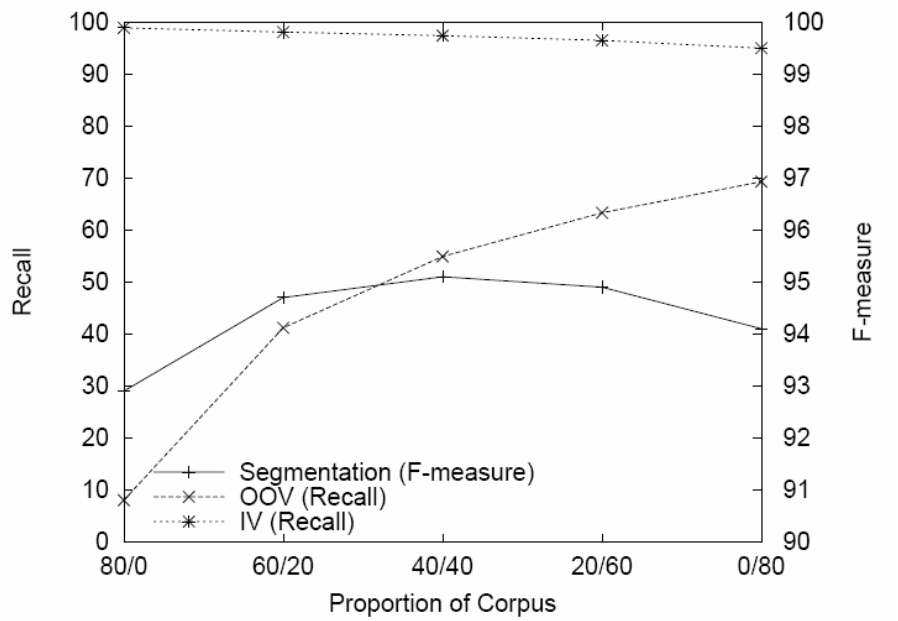

Figure 3. Accuracy of segmentation (F-measure), OOV (Recall) and IV (Recall)

4.2 Experiment with SIGHAN Bakeoff Data

As far as we know, there is no standard definition of Chinese word segmentation. A text can be segmented differently depending on the linguists who decide on the rules and also the purpose of segmentation. Therefore, it is always difficult to compare the results obtained with different methods as the data used is different. The First International Chinese Word Segmentation Bakeoff [Sproat and Emerson 2003] intended to evaluate the accuracy of different segmenters by standardizing the training and testing data. In their closed test, only the training data were used for training and no other material. Under this strict condition, it is possible to create a lexicon from the training data, but, of course, unknown words will exist in the testing data. We conducted an experiment using the bakeoff data. Since our system works only on two-byte coding, some ascii code in the data, especially numbers and letters, are converted to GB code or Big5 code prior to processing. The obtained distribution of the data is shown in Table 5. The original dictionaries consisted of all the words extracted from the training data. Some of the unknown words automatically became known words after ascii code was converted to GB/Big5 code. The conversion step reduced the number of unknown words. For example, if the number "１９９８" written in GB code existed in the training data but it was written in ascii code as "1998" in the testing data, then it was treated as an unknown word at the first location. Following conversion, it became a known word.

Table 5. Bakeoff data

Corpus	# of train words	# of test words	Unknown word rate	Size of original dictionary	Size of dictionary used
PKU	1.1M	17,194	6.9%	55,226	36,830
CHTB	250K	39,922	18.1%	19,730	12,274
AS	5.8M	11,985	2.2%	146,226	100,161
HK	240K	34,955	7.1%	23,747	17,207

The experimental setup was similar to that in Experiment 3 above. In Experiment 3, based on our previous experiments, using half of the training corpus to create the dictionary generated the best F-measure result. Therefore, only about 50% (first half) of the training corpora were used to create the dictionaries[3]. As a result, the new dictionaries contained fewer entries than the original dictionaries. Table 5 shows the details for the setting.

Table 6. Segmentation results obtained with bakeoff data

Corpus	*Recall*	*Precision*	*F-measure*	$Recall_{unknown}$	$Recall_{known}$
PKU	95.5	94.1	94.7	71.0	97.3
CHTB	86.0	83.5	84.7	57.7	92.2
HK	95.4	92.1	93.7	65.5	97.7
AS	97.0	94.8	95.9	69.0	97.6

As observed in [Sproat and Emerson 2003], none of the participants of the bakeoff could get the best results for all four tracks. Therefore, it is quite difficult to compare accuracy across different methods. Our results are shown in Table 6. Comparing with the bakeoff results, one can see that our results are not the best, but they are among the top three best results, as shown at the top of Figure 4. During the bakeoff, only two participants took part in all four tracks in the closed test. We obtained better results than one of them [Asahara *et al.* 2003], where a similar method was used to re-assign word boundaries. The difference is that words are first categorized into 5 or 10 classes (which are assumed to be equivalent to POS tags) using the Baum-Welch algorithm, and then the sentence is segmented into word sequences using a Hidden Markov Model-based segmenter. Finally, the same Support Vector Machine-based chunker is trained to correct the errors made by the segmenter. Our method which simply uses a forward and backward Maximum Matching algorithm, achieved better results than theirs when complicated statistics-based models were involved. On the other hand, compare to the results obtained by [Zhang *et al*. 2003], we only obtained better results for two

[3] Since the size of the training data is too big for the AS dataset, we had difficulty training the SVM as the time required was extremely long. Therefore, we divided it into five classifiers and finally combined the results through simple voting.

datasets and worse results for the other two datasets. They used hierarchical Hidden Markov Models to segment and POS tag the text. Although it was a closed test, they used extra information, such as class-based segmentation and role-based tagging models [Zhang *et al.* 2002], which gave better results for unknown word recognition. The bottom of Figure 4 shows the results of unknown word detection. Again, our method performed comparatively well in detecting unknown words.

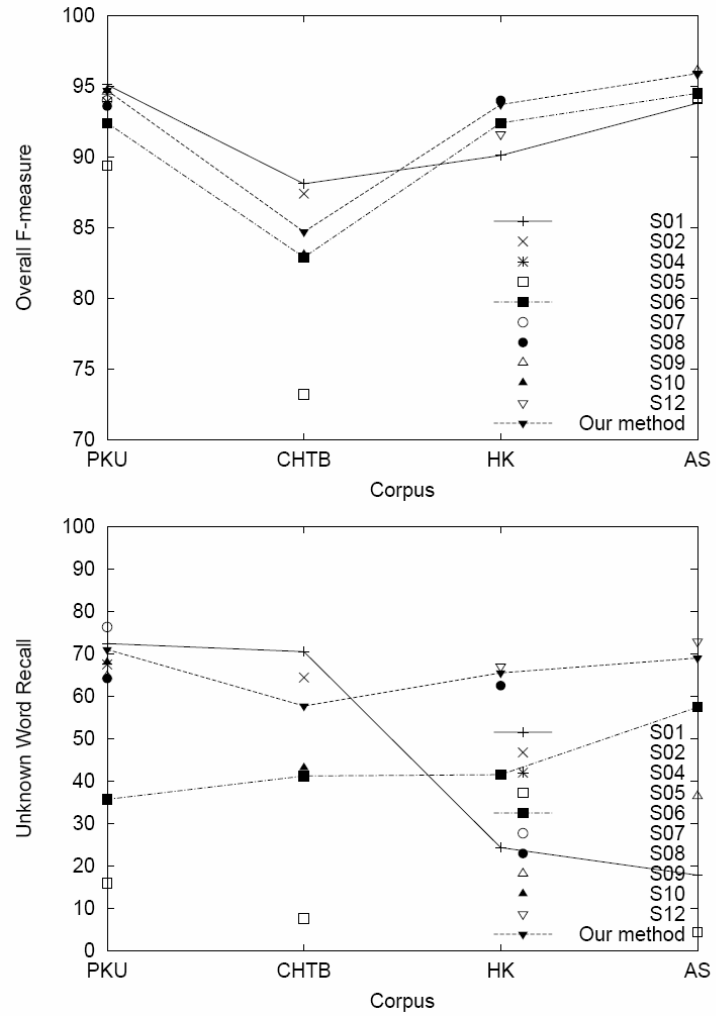

Figure 4. Comparision of bakeoff results (overall F-measure and unknown word recall)

Regarding Chinese word segmentation problem as character tagging problem has previously been seen in [Xue and Converse 2002]. The difference in our method is that we supply FMM and BMM outputs as a control for the final output decision. However, only

words from half of the training corpus are controlled. Since false unknown words are the main cause of errors with known words, our method tries to maintain accuracy for known words while at the same time detecting new words. As Xue and Converse [2002] used a different corpus than ours, namely, the Penn Chinese Treebank, it is difficult to make a fair comparison. They also participated in the bakeoff for the HK and AS tracks only [Xue and Shen 2003]. They obtained segmentation F-measures of 91.6 and 95.9, respectively, while we achieved 93.7 and 95.9, which are quite comparable. They did a bit better in unknown word recall, achieving 67.0% and 72.9% recall rates, whereas ours were 65.5% and 69.0%. On the other hand, we obtained much better results in known word recall, 97.7% and 97.6%, compared to their recall rates of 93.6% and 96.6%. Usually a piece of text contains more known words than unknown words; therefore our method, which controls the outputs of known words, is a correct choice. Furthermore, our method can also detect unknown words with comparable results.

In conclusion, our results did not surpass the best results in the bakeoff for all datasets. However, our method is simpler. We only need a dictionary that can be created from a segmented corpus, FMM and BMM modules, and a classifier, without the use of human knowledge. We can get quite comparable results for both known words and unknown words. The results are worse when the training corpus is small and there exist a lot of unknown words, such as in CHTB testing data. Therefore, we still need to investigate the relationship between the size of the training corpora and the proportion of the corpora used to create the dictionaries in the training for solving ambiguity problems and performing unknown word detection. We are also looking into the possibility of designing an ideal model, where optimal results for known words, as in Experiment 2, and unknown words, as in Experiment 4, can be obtained.

5. Conclusion

Our proposed method generated better results than the baseline models, namely, FMM and BMM. We achieved nearly 99% recall when unknown words did not exist. However, in the real world, unknown words always exist in texts, even if we use a very large dictionary. Therefore, we also embed a model to detect unknown words. Unfortunately, while the accuracy achieved in unknown word detection increases, the performance in solving the known word ambiguity problem declines. As shown by the experiments on the bakeoff data, our model works well only when the training corpus is large. In conclusion, while our model is suitable for solving the segmentation ambiguity problem, it can also perform unknown word detection at the same time. However we still need to find a balance that will enable us to solve these two problems optimally. We also need to research the relationship between the training corpus size and the best proportion of the corpus used to create the dictionary for training to solve the ambiguity problem and perform unknown word detection.

Acknowledgements

Thanks go to Mr. Kudo for his Support Vector Machine-based chunker tool, *Yamcha*. We also thank Peking University and SIGHAN for providing the corpora used in our experiments. Finally, we thank the reviewers for their invaluable and insightful comments.

Reference

Asahara, M., C.L. Goh, X.J. Wang and Y. Matsumoto, "Combining Segmenter and Chunker for Chinese Word Segmentation," In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, 2003, pp. 144–147.

Chen, K.J. and M.H. Bai, "Unknown Word Detection for Chinese By a Corpus-based Learning Method," In *Proceedings of ROCLING X*, 1997, pp. 159–174.

Chen, K.J. and W.Y. Ma, "Unknown Word Extraction for Chinese Documents," In *Proceedings of COLING 2002*, 2002, pp. 169–175.

Fu, G.H. and K.K. Luke, "An Integrated Approach for Chinese Word Segmentation," In *Proceedings of PACLIC 17*, 2003, pp. 80–87.

Fu, G.H. and X.L. Wang, "Unsupervised Chinese Word Segmentation and Unknown Word Identification," In *Proceedings of NLPRS*, 1999, pp. 32–37.

Huang, C.N., "Segmentation Problem in Chinese Processing," *Applied Linguistics*, 1, 1997, pp. 72–78.

Kudo, T. and Y. Matsumoto, "Chunking with Support Vector Machines," In *Proceedings of NAACL*, 2001, pp. 192–199.

Lai, Y.S. and C.H. Wu, "Unknown Word and Phrase Extraction Using a Phrase-Like-Unit-Based Likelihood Ratio," In *Proceeding of ICCPOL '99*, 1999, pp. 5–9.

Li, M., J.F. Gao, C.N. Huang and J.F. Li, "Unsupervised Training for Overlapping Ambiguity Resolution in Chinese Word Segmentation," In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, 2003, pp. 1–7.

Luo, X., M.S. Sun and B. K. Tsou, "Covering Ambiguity Resolution in Chinese Word Segmentation Based on Contextual Information," In *Proceedings of COLING 2002*, 2002, pp. 598–604.

Ma, W.Y. and K.J. Chen, "A Bottom-up Merging Algorithm for Chinese Unknown Word Extraction," In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, 2003, pages 31–38.

Nie, J.Y., M.-L. Hannan and W.Y. Jin, "Unknown Word Detection and Segmentation of Chinese Using Statistical and Heuristic Knowledge," *Communications of COLIPS*, 5, 1995, pp. 47–57.

Sang, E. F.-T.K. and J. Veenstra, "Representing Text Chunks," In *Proceedings of EACL '99*, 1999, pp. 173–179.

Shen, D.Y., M.S. Sun, and C.N. Huang, "The application & implementation of local statistics in Chinese unknown word identification," *Communications of COLIPS*, 8(1), 1998, pp. 119–128.

Sproat, R. and T. Emerson, "The First International Chinese Word Segmentation Bakeoff," In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, 2003, pp. 133–143.

Uchimoto, K., Q. Ma, M. Murata, H. Ozaku and H. Isahara, "Named Entity Extraction Based on A Maximum Entropy Model and Transformational Rules," In *Processing of the ACL 2000*, 2000, pp. 326–335.

Vapnik, V. N., *The Nature of Statistical Learning Theory*, Springer, 1995.

Xue, N.W. and S. P. Converse, "Combining Classifiers for Chinese Word Segmentation," In *Proceedings of First SIGHAN Workshop on Chinese Language Processing*, 2002, pp. 57–63.

Xue, N.W. and L.B. Shen, "Chinese Word Segmentation as LMR Tagging," In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, 2003, pp. 176–179.

Zhang, H.P., Q. Liu, H. Zhang and X.Q. Cheng, "Automatic Recognition of Chinese Unknown Words Based on Roles Tagging," In *Proceedings of First SIGHAN Workshop on Chinese Language Processing,* 2002, pp. 71-77.

Zhang, H.P., H.K. Yu, D.Y. Xiong and Q. Liu, "HHMM-based Chinese Lexical Analyzer ICTCLAS," In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, 2003, pp. 184–187.

Zhou, G.D. and K.T. Lua, "Detection of Unknown Chinese Words Using a Hybrid Approach," *Computer Processing of Oriental Language*, 11(1), 1997, pp. 63–75.

Computational Linguistics and Chinese Language Processing
Vol. 10, No. 3, September 2005, pp. 397-416

The Design and Construction of the PolyU Shallow Treebank

Ruifeng Xu*, Qin Lu*, Yin Li* and Wanyin Li*

Abstract

This paper presents the design and construction of the PolyU Treebank, a manually annotated Chinese shallow treebank. The PolyU Treebank is based on shallow annotation where only partial syntactical structures within sentences are annotated. Guided by the Phrase-Standard Grammar proposed by Peking University, the PolyU Treebank has been designed and constructed to provide a large amount of annotated data containing shallow syntactical information and limited semantic information for use in natural language processing (NLP) research. This paper describes the relevant design principles, annotation guidelines, and implementation issues, including the achievement of high quality annotation through the use of well-designed annotation workflow and effective post-annotation checking tools. Currently, the PolyU Treebank consists of a one-million-word annotated corpus and has been used in a number of NLP research projects with promising results.

Keywords: Shallow Treebank, Shallow Parsing, Corpus Annotation, Natural Language Processing

1. Introduction

A treebank can be defined as a syntactically processed corpus. It is a language resource with linguistic information annotated at, variously, the word, phrase, clause, and sentence levels, in order to form a bank of linguistic trees. Many treebanks have been constructed for different languages, including Penn Treebank [Marcus *et al.* 1993] and the ICE-GB [Wallis *et al.* 2003] for English, and the Penn Chinese Treebank [Xia *et al*. 2000; Xue *et al.* 2002] and the Sinica Treebank [Chen *et al*. 1999; Chen *et al.* 2003] for Chinese.

Most of the reported Chinese treebanks, including the Penn Chinese Treebank and Sinica Treebank, are based on full parsing, where complete syntactical analysis is performed. This includes determining the syntactic categories of words, locating chunks that can be nested,

* Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
Tel: +852-27667326; +852-27667247; +852-27667325 Fax: +852-27740842
E-mail: {csrfxu, csluqin, csyinli, cswyli}@comp.polyu.edu.hk

[Received October 13, 2004; Revised March 18, 2005; Accepted March 21, 2005]

finding relations between phrases, and resolving attachment ambiguities. Thus, the output of full parsing is a set of complete syntactic trees. Due to the complexity of natural languages, automatic full parsing is still quite challenging. An alternative to automatic full parsing is to adopt a divide-and-conquer strategy, i.e., to divide full parsing into several independent sub-tasks which can be applied relatively easily. One of these sub-tasks is shallow (or partial) parsing. The purpose of shallow parsing is to identify local syntactical structures that are relatively simple and easy to identify while ignoring the complicated task of analyzing how these phrases are syntactically used to construct sentences. Thus, shallow parsing only identifies local structures in sentences. These local structures form the sub-trees of a full syntactic tree. Because shallow parsing does not involve complex and ambiguous attachment analysis, it can find some local structures at much lower cost and with a much higher accuracy. For these reasons, shallow parsing has in recent years been the focus of more research, and it has been applied in many NLP applications. However, the lack of a large-scale Chinese shallow treebank has been an impediment to research in this area. This has motivated us to construct a Chinese shallow treebank for Chinese natural language processing applications. This treebank, referred as the PolyU Treebank, is named after the University where it is being developed.

One problem with shallow parsing is that, unlike full parsing, it seeks to identify only certain local structures in a sentence. Furthermore, at present, there is no widely-accepted common standard for the determining scope and depth of local structures, and different reported works vary in how they define what local structures are [Dalemans *et al.* 1999; Sun 2001; Li *et al.* 2003]. Therefore, in this work, we will first discuss the objectives of shallow parsing based on our needs and those of other NLP researchers and define the scope of shallow parsing. In accordance with this defined scope, we will then show how the PolyU Treebank has been constructed by manually annotating shallow syntactic structures from a selected corpus.

Obviously, the scope and the depth of shallow annotation should be determined based on the requirements of the applications using the treebank. Based on the typical requirements of NLP research tasks such as Chinese collocation extraction, terminology extraction, and the acquisition of descriptions of terminologies conducted at the authors' research institution, we restrict shallow syntactic structures to the *maximal phrases* that play various roles as subjects, predicates, complement clauses and other syntactic components in sentences. Within the scope of the present work, our aim is to identify *base-phrases*, that is minimum syntactic unit in a maximal phrase. We also identify those nested phrases between base-phrases and maximal phrases which we call *mid-phrases*. Maximal phrase, Base-phrase, Mid-phrase will be defined in detail in Section 3. Each identified phrase is given a mandatory syntactic label and an optional semantic label. Its header is also identified. An important feature of our treebank is

that the identified phrases are augmented with semantic information. This kind of information is useful in many areas of NLP research but is difficult to identify automatically and sometimes not annotated in the other existing treebanks.

For guidance in syntactic annotation, we choose to use the Phrase-Standard Grammar (PSG) as proposed by Peking University [Yu *et al.* 1998]. There are two reasons for this choice. First, the PSG grammar framework is widely accepted in mainland China. Second, in order to reduce the cost of annotation and to ensure the maximum sharing of our output, we perform shallow syntactic annotation on the segmented and tagged People's Daily corpus, developed in Peking University [Yu *et al.* 2001].

The process of constructing our treebank, which has taken more than 15 months, has included guideline design, the development of annotation specifications, and annotation and quality assurance checking. The one-million-word annotated shallow treebank is more than 98.8% accurate in terms of phrase bracketing and more than 98% accurate in phrase labeling. Such a large-scale treebank can be used to support a variety of NLP research. Currently, it has been used to train and to test a shallow parser [Lu *et al.* 2003]. Furthermore, other research conducted in authors' institution, including Chinese collocation extraction, Chinese terminologies extraction, and information retrieval, have also benefited from the PolyU Treebank. We are currently optimizing the treebank and making it available to other researchers as a public resource.

This paper presents the major issues involved in the design and construction of the PolyU Treebank and its quality control mechanisms. The rest of this paper is organized as follows. Section 2 introduces the design principles. Section 3 describes the annotation guidelines. Section 4 describes the tasks involved in annotating the PolyU Treebank, including corpus data preparation, word segmentation, POS tagging, phrase bracketing, and phrase labeling specifications. Section 5 discusses the quality assurance mechanisms and the post-annotation checking tools developed for this project. Section 6 gives some examples to illustrate how this shallow treebank can be used in NLP. Section 7 gives conclusions.

2. Design Principles

Due to the fact that currently, no large-scale shallow-annotated Chinese treebanks are available, in the course of designing PolyU Treebank, we referenced two important fully-annotated Chinese treebank: the Penn Chinese Treebank and the Sinica Treebank. The Penn Chinese Treebank was annotated based on the Government and Bind framework and contains about 500,000 Chinese words, most of which were mainly manually annotated according to a strict quality assurance process [Xue *et al.* 2002]. The Sinica Treebank was developed by the Academic Sinica, Taiwan. Phrase bracketing and annotation were carried out using a head-driven chart parser guided by Information-based Case Grammar (ICG), and

followed by manual post-editing. The Sinica Treebank contains 39,000 parsed trees and 329,000 words [Chen *et al*. 1999; Chen *et al.* 2003]. A natural way to obtain a shallow treebank is to extract shallow structures from a fully annotated treebank. Unfortunately, the Penn Treebank and Sinica Treebank were annotated using different grammar frameworks as well as different word segmentation/POS tagging strategies, making them unsuitable for our annotation scheme.

To ensure that the PolyU Treebank would be high in quality and widely accepted, it was designed and constructed based on four basic principles:

Principle 1: High resource-sharing capability

The PolyU Treebank was designed to sever as a general purpose treebank for use in as wide a range of applications as possible. This called for the selection of an effective and well-accepted grammatical framework for representing syntactical information as well as for a well-accepted word segmentation/POS tagging scheme.

We chose to use the Phrase-Standard Grammar (PSG), proposed by Peking University. PSG is widely accepted by Chinese NLP researchers. In the PSG framework, phrases rather than words are treated as basic Chinese syntactical units. The reason is that while an individual word can be used in different ways and may have different part-of-speech (POS) tags representing its different functions in sentences, a phrase is made up of a number of words normally driven by a headword, and consequently, has a stable internal structure and order. Based on this framework, syntactical analysis should be performed in a cascaded fashion, and a linear character string can finally be syntactically analyzed to form a cascaded tree.

In the absence of an orthographic device for delimiting words in Chinese, it is necessary to segment words before performing POS tagging. We used a segmented and tagged corpus consisting sentences from the People's Daily, annotated by Peking University. This corpus was accurately segmented and tagged in accordance with the PSG framework, and contains articles from the People's Daily published in 1998. The claimed accuracy of word segmentation and POS tagging is 99.9% and 99.5%, respectively [Yu *et al.* 2001]. Using this popular and accurate resource significantly reduced the cost of annotation in our research and ensured the maximum sharing of our output.

Principle 2: Low structural complexity

The second design principle was that the PolyU Treebank should not be structurally very complex; its annotation framework should be clear and simple and its syntactic and functional information should be labeled according to commonly used and widely accepted standards.

To ensure that our shallow annotation approach satisfied the requirements typical language applications in terms of syntactical information, we chose to focus on the annotation

of phrases and the identification of headwords while ignoring sentence-level syntax. More specifically, we wanted to identify three types of information: (1) *base-phrases,* that is, non-nesting phrases with at least one headword; (2) *maximal phrases,* that is, phrases that marked the boundary of our scope of examination, inclosing the base-phrases and plays the role of subject, predicate, complement clause, embedded clause, or other syntactic components of sentences; and (3) *mid-phrases,* that is the intermediate nesting phrases between base-phrases and maximal phrases if they existed. Maximal phrases and base-phrases will be defined and discussed in detail in Section 3. As for mid-phrases, a limit was imposed on the level of nesting since we did not intend to provide full parsing information. In order to limit the structural complexity, we limited nesting brackets to only three levels. In other words, mid-phrases were limited to only at most one level.

Principle 3: Sufficient and useful syntactic information

The third design principle was to provide syntactic information at a low level of complexity that would be useful for and effective in a wide variety of NLP applications. Earlier works in Chinese shallow annotation had annotated only non-nesting base-phrases [Sun 2001]. However, base-phrase annotation alone is not adequate for many applications. Our annotation scheme permits three levels of nesting, and this has a number of advantages. First, maximal phrases indicate the essential syntactic elements of a sentence, such as the subject and predicate, and the availability of this information makes it is possible in many applications to refine the search context window. Secondly, base-phrases are the simplest and most stable structural elements of a sentence. Thus, they are regarded as the smallest syntactic units. Lastly, nested mid-phrases are useful for describing distant modifier relations within maximal phrases, which is helpful in certain applications.

The PolyU Treebank provides not only adequate syntactical information but also some semantic information. To achieve this, each phrase is given a syntactic label and sometimes also a label providing semantic information. For example, *"国家航空和宇宙航行局"(NASA)* is a noun phrase and is assigned the label *NP*. Furthermore, in terms of semantics, it is a noun phrase that indicates the name of an organization, so it is given the appropriate additional label, *NT*. The fact that the PolyU Treebank is a "Not-So-Shallow" treebank makes it substantially different from and more useful than other base-phrase only shallow treebanks. The information it provides can be used in language applications to remove ambiguities. Finally, we should point out that in our treebank, the headword of a base-phrase is also annotated.

Principle 4: Large quantities of annotated data with great accuracy

The sizes of existing Chinese treebanks range from 100,000 to 500,000 words. It is an acceptable size for full parsing [Leech and Garside 1996] but not sufficient for lexical-level analysis. With reference to work on the English language, it is our goal to create a treebank of

one million words. A treebank of this size can support the design and training of a shallow parser and be directly used in the collocation extraction and named entity identification work being conducted by authors' research group.

A well-developed treebank must be very accurately annotated. With the goal of reducing annotation errors, we have designed clear and simple annotation guidelines. To avoid inaccuracies arising from automatic parsing, we have performed annotation manually, and post-annotation error and consistency checking have been performed with tools developed by us. Finally, to avoid human errors, some texts are double- and triple-annotated and then compared. This allows makes it easy to identify and correct errors.

3. Annotation Guideline Design

The establishment of annotation guidelines is the first step in treebank development. To ensure high quality output, the guidelines must follow the design principles and must be clear, unambiguous, easy to understand, and easy to follow. The PolyU Treebank guidelines include definitions of (1) syntactical phrase categories, (2) categories of semantic information, and (3) different phrase levels, including maximal phrases, mid-phrases and base-phrases. Because the PolyU Treebank is based on a segmented and POS tagged corpus, the part-of-speech tags in the corpus are used (with only minor modifications for the sake of annotation consistency). Appendix 1 provides a complete list and explanations of the POS tags. These tags will be used in the examples provided in this paper.

Brackets, [and] are used to indicate the left and right boundaries of phrases. The right bracket is appended with syntactic labels in the form of *[Phrase]SS-FF*, where *SS* is a mandatory syntactic label, such as *NP*(noun phrase) and *AP*(adjective phrase), and *FF* is an optional label indicating internal semantic information, such as *BL*(parallel). For example, a noun phrase with parallel components will be annotated as *[荣誉/n 与/c 尊严/n]NP-BL (honor and dignity)*.

3.1 Defining the syntactical phrase categories

The first level of information for describing phrases is that in the syntactical phrase category. With reference to the works of Penn Chinese Treebank and Sinica Treebank, our guidelines define a total of eight syntactical phrase categories:

NP — Noun phrase. An *NP* is headed by a noun and the header is normally the last noun in the phrase, e.g., *[市场/n 经济/n]NP (market economy).*

TP — Time phrase. A *TP* consists of continuous time words and is used to indicate a time, e.g., *[早上/t 8 时/t]TP (8:00 in the morning).*

FP — Position phrase. A *FP* is headed by a position word, *f*, and is used to indicate position information, e.g., *[内蒙古/ns东北部/f#]FP (North-east of Inner Mongolia).*

VP — Verb phrase. A *VP* is a phrase headed by a predicate and containing no subject, e.g., *[顺利/a启动/v#]VP-ZZ (successfully start), and [分析/v# 问题/n]VP-SBI (analyze the problem).*

AP — Adjective phrase. The header of an *AP* is an adjective and the whole phrase acts as an adjective in the sentence, e.g.,*[公正/a合理/a#]AP (fair and reasonable).*

DP — Adverb phrase. The header of a *DP* is an adverb, and the whole phrase plays the role of an adverbial role in a sentence, e.g., *[已/d 不再/d#]DP (no longer).*

PP — Preposition phrase. A *PP* is the phrase which begins with a preposition, e.g., *[在/p贵州/ /ns农村/n]PP (In the countryside of Guizhou Province).*

QP — Quantifier phrase. A *QP* consists of a number and a quantifier. The quantifier acts as the header. Normally, a *QP* is used as the modifier of an *NP* or a *VP*, e.g., *[[数千/m名/q#]QP 士兵/n (several thousand soldiers).*

3.2 Defining semantic information categories

The PolyU Treebank is unique in that it is annotated with semantic labels. A annotation of the *FF* labels is not mandatory. Only those phrases with pre-defined semantic phrase categories are labeled. Semantic information is very useful for some language applications. For example, *山东/ns 烟台/ns 市/n (Yantai City, Shan Dong Province)* and *烟台/ns 大学/n (Yantai University)* are both noun phrases, but the first one is the name of a place and the second that of an organization. Using the semantic information labels *NS* (Name of a place) and *NT* (Name of an organization) allows one to distinguish between these two NPs. This is highly useful in named entity extraction and automatic summarization. The additional semantic labels can be considered a natural byproduct of manual annotation since annotators naturally need to go through the mental process of identifying them. We simply making them available so that such used knowledge are not wasted during annotation.

In the following, we listed the semantic categories.

Semantic information categories for Noun Phrases

NT — Name of an organization, e.g., *[烟台/ns 大学/n]NP-NT (Yantai University).*

NS — Name of a place, e.g., *[江苏省/ns铜山县/ns]NP-NS (Jiangsu Province, Tongshan Country).*

NR — Name of a person, e.g., *[胡/nr 锦涛/nr]NP-NR (Hu Jintao).*

NZ — Other proper noun phrase, e.g., *[诺贝尔/nr奖/n]NP-NZ (The Nobel Prize).*

BL — Juxtaposition structure. A *BL* label indicates that the phrase is made up of two or more parallel components, e.g., *[中国/ns 与/c 南非/ns]NP-BL (China and South Africa).*

FZ — Appositive. An *NP* with *FZ* labels normally has two equivalents, e.g., *[[国家/n主席/n]NP [江/nr 泽民/nr]NR]NP-FZ (the president of China, Jiang Zemin).*

PZ — Noun modifier. A *PZ* is the default semantic structure of an *NP,* e.g., *[美丽/a 的/u 花/n#]NP-PZ (beautiful flower).*

FS — Noun plurals. A *FS* indicates that the last word in a noun phrase is a suffix for noun plurals, e.g., *[朋友/j# 们/k]NP-FS (friends).*

DE — A *DE* construction is a special kind of an *NP* structure in Chinese. It ends with "的"(*DE*) and indicates the absence of the complementation, e.g., *比/v[原先/d预料/v的/u]NP-DE低/a (lower than originally expected).*

SU — A *SU* construction is a special kind of *NP* structure in Chinese. The typical pattern is *所(SUO)+VP+NP,* e.g., *[所/u画/v禽鸟/n#]NP-SU (the birds painted by).*

Semantic information categories for Verb Phrases

SBI — Predicate and its object. A *VP* with the label *SBI* contains of a predicate and an object, e.g., *[打/v# 篮球/n]VP-SBI 是/v 我/r 的/u 爱好/n (playing basketball is my hobby).*

SBU — Complement. The label *SBU* indicates that the second part of the *VP* phrase is the complement modifying the first part of the *VP*, e.g.*[医治/v# 无效/v]VP-SBU (ineffectively treat).*

ZZ — When a *VP* has the label *ZZ*, the verb is the header and other words are its modifiers, e.g., *[[有效/ad 打击/v#]VP-ZZ了/u 敌人/n]VP-SBI (effectively strike the enemy).*

SD — Serial verb constructions. A *SD* indicates that there are serial actions in a *VP* phrase, where the last action is the cardinal action, e.g., *[[审核/v 发放/v]VP-SD 护照/n]VP-SBI (verify and issue the passport).*

BA — A *BA* construction is a special kind of *VP* structure in Chinese. The typical pattern is*把(BA)+NP1 +VP*, e.g., *[把/p[扶贫/vn开发/vn工作/vn]NP-PZ 作为/v#]VP-BA (place the work of poverty reduction and social development as).*

BEI — A *BEI*-construction is a special kind of a *VP* structure in Chinese. The typical patterns are*被(BEI)+ NP+VP and NP+被+VP*, e.g., *商店/n [被/p[责令/v# 停业/vn]VP-SBI]VP-BEI (the shop was ordered to close).*

Semantic information categories for Time Phrases

PO — A point-of-time indicator. The label *PO* indicates that the *TP* carries point-of-time information, e.g., *[7月/t 1日/t]TP-PO (July 1).*

DU — A period-of-time indicator. A *DU* indicates a period of time, e.g., *[今后/t 3/m年/q]TP-DU (following three years).*

Semantic information categories for Prepositional Phrases

YY — Causation information. A *YY* label is used only to modify a *PP* to indicate that the *PP* carries causation information, e.g., *[因/p 饿/a]PP-YY 死亡/v (starved to death).*

DX — Object information. The label *DX* is used to modify a *PP* to indicate object information, e.g., *[向/p [受灾/vn 地区/n]NP]PP-DX (to the disaster area).*

DD — Place information. This is the place indicator of a *PP,* e.g., *[在/p 深圳/ns]PP-DD (in Shenzhen).*

FM — Method information. A *PP* with an *FS* label signals the existence of method information, e.g., *[通过/p [股票/n 上市/v]S]PP-FM (Through the stock market).*

MD — Motivation information. A *PP* with an *MD* label signals the existence of motivation information, e.g., *[为/p 动武/v]PP-MD [找/v 借口/n]VP-SBI (looking for an excuse for war).*

GJ — Tool information. A *GJ* label indicates that a *PP* carries tool information, e.g., *[用/p 公车/n]PP-GJ (using a public-bus).*

SJ — Time information. A *SJ* label indicates that a *PP* carries time information, e.g., *[到/v 目前/t 为止/v]PP-SJ (up to now).*

3.3 Phrase bracketing

Phrases in the PolyU Treebank are divided into three levels: maximal phrases, mid-phrases and base-phrases. The syntactical analysis and annotation of the PolyU Treebank begins with the identification of maximal phrases which define the scope of examination for bracketing.

A *maximal phrase* is a predicate that plays the role a distinct syntactic component of a sentence, realized by the maximum span of its non-overlapping length. Maximal phrases form the backbone of a sentence. The identification of maximal phrases is one of the most difficult steps in the whole process in that annotators have to syntactically analyze sentences and understand their syntactic components even though they have not yet been labeled. The objective of identifying maximal phrases is to separate a sentence into several syntactic components for examination. After maximal phrases are identified, the base-phrases can then be identified within the scope of examination, that is, within each maximal phrase.

A *base-phrase* is defined as a minimum non-nesting phrase with a stable internal structure and independent semantic role. Normally, a base-phrase has a lexical word as its headword. Essentially, a base-phrase must consist of continuous words and contain no nesting components. It never overlaps with other phrases and must be contained within a maximal

phrase. Base-phrases normally conform to a number of typical patterns, such as *[a+n]->NP*, *[a+a]->AP*.

A *mid-phrase* is a nested phrase within a maximal phrase and has a base-phrase as its header. A mid-phrase may contain more than one base-phrase, but only one will be its header. A mid-phrase may have nested components, but none of them may overlap.

The headword of each phrase is also annotated. Further details and examples of phrase bracketing will be provided in Section 4.

4. Implementation of the PolyU Treebank

4.1 Corpus data preparation

The People's Daily corpus, developed by Peking University, consists of more than 13,000 articles and a total of five million words. Since only one million words are required in the PolyU Treebank, we carried out a data selection process. To avoid the duplication of short-lived events and topics, we treated each day's news as a single unit, and we picked six random days in each month from among the six months of data in the entire collection as the raw treebank data.

4.2 Word Segmentation and Part-of-Speech Tagging

In the tasks of the word segmentation and POS tagging of the People's Daily corpus, we were guided by the PSG grammar and "The Grammatical Knowledge-base of Contemporary Chinese" [Yu *et al.* 1998]. The specifications include a total of 43 POS tags. Peking University claimed that the accuracy of word segmentation and POS tagging was higher than 99.9% and 99.5%, respectively [Yu *et al.* 2001].

In this project, we directly used the PKU POS tagging results and made only some notational changes. These changes were made to ensure consistent labeling in our system, where lower cases are used to in word-level tags and upper cases are used in phrase-level labels.

4.3 Phrase Bracketing and Annotation

Identification of Maximal-phrases:

A maximal phrase contains at least one base-phrase and plays a syntactic role in the sentence. Consider the following example sentence:

中国/ns 旅游年/n 是/v 一/m 次/q 国家级/b 的/u 宣传/vn 促销/vn 活动/vn (Example.1)

(China Tourism Year is a national-level promotion and marketing activity)

We find that the above sentence has a S-V-O structure. *中国/ns 旅游年/n* is the subject, *是/v* is the predicate, and *一/m 次/q 国家级/b 的/u 宣传/vn 促销/vn 活动/vn* is the object. Clearly there are three syntactic components in this sentence, thus, two separate maximal-phrases, *[中国/ns 旅游年/n]NP (China Tourism Year)* and *[一/m 次/q 国家级/b 的/u 宣传/vn 促销/vn 活动/vn]NP (a national-level promotion and marketing activity)* are annotated. Note that *是/v* is also considered a maximal phrase because it acts as a predicate. However, since it has only one lexical word and is structurally unambiguous, by default, it is not bracketed. Admittedly, *是/v* and *一/m 次/q 国家级/b 的/u 宣传/vn 促销/vn 活动/vn* can be constructed as a *VP*, but we regard this kind of bracketing is more useful for indicating how phrases may be used to construct a sentence. That is to say, this kind of bracketing would take us into the realm of full parsing, which is not our objective. Thus, we choose to bracket them as separate phrases. As a result, the maximal phrase annotation result is

> *[中国/ns 旅游年/n]NP 是/v [一/m 次/q 国家级/b 的/u 宣传/vn 促销/vn 活动/vn]NP-PZ.*

Consider another example,

> *富裕/v 起来/v 的/u 当地/a 农民/n 自发/d 地/u 组织/v 了/u 多个/a 业余/a 乐团/n*
>
> (the rich farmers took the initiative to organize several amateur bands) (Example 2)

We can separate this sentence into three components, *富裕/v 起来/v 的/u 当地/a 农民/n* is the subject, *自发/d 地/u 组织/v 了/u* is the predicate, and *多个/a 业余/a 乐团/n* is the object. Thus, this sentence is annotated with three maximal phrases, bracketed and labeled as follows:

> *[富裕/v 起来/v 的/u 当地/a 农民/n#]NP [自发/d 地/u 组织/v# 了/u]VP-ZZ [多个/a 业余/a 乐团/n]NP-PZ*

Most syntactical labels can be used in maximal phrases, except for *AP* (adjective phrase), *DP* (adverb phrase), and *QP* (quantifier phrase). Meanwhile, *NP-NT*, *NT-NS*, *NP-NZ* may only be used to label maximal phrases. These types of phrases do not normally contain nesting components or header words.

Base-phrases Identification:

Base-phrases are identified only within an already-identified maximal phrase, either nesting inside it or overlapping it. Normally a base-phrase contains two-to-four words with one lexical word as its header.

Take the maximal phrase *[一/m 次/q 国家级/b 的/u 宣传/vn 促销/vn 活动/vn]NP-PZ* in **Example 1** as an example, *[一/m 次/q]QP (a) and [宣传/vn 促销/vn 活动/vn#]NP-PZ (promotion and marketing activity)* are base-phrases in this maximal phrase. Thus, the sentence is annotated as follows:

[中国/ns 旅游年/n]NP 是/v [[一/m 次/q]QP 国家级/b 的/u [宣传/vn 促销/vn 活动/vn]NP-PZ]NP-PZ.

As it happens, *[中国/ns 旅游年/n]NP* and *是/v* are also base-phrases, but because they overlap with maximal phrases, they are not further bracketed. Our annotation principle here is that if a base-phrase overlaps with a maximal phrase, it will not be bracketed twice.

It should be pointed out that the identification of base-phrase is the most fundamental and important goal of treebank annotation. The identification of maximal phrases can be thought as the parsing of a clause using a top-down approach. The identification of base-phrase is however, follows bottom-up approach, the object of which is to identify the most basic units within maximal phrases.

Mid-Phrases Identification:

Because other syntactic structures may sometimes exist between base-phrases and maximal phrases, it is useful to identify one more level of syntactic structure within a maximal-phrase, the mid-phrase. This step begins with the examination of a base-phrase. Thus, **Example 1** is further annotated as follows:

[中国/ns 旅游年/n]NP 是/v [[一/m 次/q]QP [国家级/b 的/u [宣传/vn 促销/vn 活动/vn]NP-PZ]NP-PZ]NP-PZ

where, the underlined text contains the additional annotations.

As we limit nesting to three levels, any further nested phrases are ignored. The following sentence shows the result of annotation with three levels of nesting:

[目前/t [企业/n 发展/vn]NP [值得/v 注意/v 的/u [[几/m 个/q]QP 问题/n]NP-PZ]NP]NP

(s*everal issues which are worthy of consideration in the development of current enterprise*).

Full annotation would identify four levels of nesting, as shown below, but our system does not include the additional level of bracketing indicated by the underlined annotations as this is beyond our limit of 3 levels.

[目前/t [[企业/n 发展/vn]NP [值得/v 注意/v 的/u [[几/m 个/q]QP 问题/n]NP-PZ]NP]NP]NP.

Annotation of Headwords

In our system, a '#' tag is appended to a word to indicate that it is a headword. Here, a headword must be a lexical word (sometimes also called a content word) rather than a function word. In most cases, a headword stays in a fixed position in a base-phrase. For example, the headword of a noun phrase is normally the last noun in the phrase. Thus, it is considered to be in the default position and to need no explicit annotation. For example, in the clause

[美国/ns 科学家/n]NP [绘制/v 出/v]VP-SBU (t*he American scientists drafted*),

[绘制/v 出/v] (*drafted*) is a verb phrase, and the headword of the phrase is *绘制/v*, which is not in the default position for a verb phrase headword. Thus, this phrase is further annotated as: *[美国/ns 科学家/n]NP [绘制/v# 出/v]VP-SBU*. Note that *科学家/n* is also a headword in *[美国/ns 科学家/n] (the American scientist*s), but since it is in the default position (for the noun phrase *NP*, according to the default grammatical structure, the last noun in the phrase is the headword, and the other components are the modifiers taking the *PZ* label), no explicit annotation is needed.

5. Quality Assurance and Annotation Progress

Our research team is made up of four people from the Hong Kong Polytechnic University (HKPU), two linguists from Beijing Language and Culture University (BLCU), and some research collaborators from Peking University. The annotation work has been carried out by four post-graduate students of languages and computational linguistics from BLCU.

5.1 Quality Assurance

To achieve high quality annotation, guidelines and annotation specifications must be carefully prepared. In the first stage, two linguists from China worked with the team in Hong Kong to prepare annotation guidelines. At this stage, the annotation range of syntactic categories and semantic information categories were also determined. Then, sample annotation was performed in Hong Kong, and the results were summarized to identify some typical patterns for constructing phrases. After that, all the members annotated in duplicates a 60,000-word sample according to the draft specifications. Based on analysis of the results and feedback, the specifications were revised.

In the annotation stage, about 25% of the materials were distributed in identical form to the annotators. When the first pass annotation was finished, the duplicate annotations were compared. Inconsistencies were discussed to identify the most appropriate annotation results. This result was then taken as the ultimate standard (the so called *Gold Standard)* for evaluating inter-annotator accuracy and consistency. The annotators were required to study this Gold Standard and to use it as the basis for correcting mistakes in their own annotations.

Furthermore, a group of checking and evaluating tools were developed. The first tool performs post-annotation checking to ensure that (1) all Part-of-Speech tags are valid, (2) all phrase boundary marks are matched, (3) there are no cross-bracketed phrases, and (4) all the phrase syntactical labels and semantic labels are annotated in the correct format. This tool is effective for removing obvious annotation mistakes.

The most difficult task is to maintain inter-annotator consistency. To assist this work, we developed two tools. A multiple annotation checking tool was developed to compare and evaluate duplicate annotation results. Any mismatches in phrase brackets and labels were detected and manually verified using the tool. Such annotation error cases were used to train the annotators so that they could then manually remove similar annotation errors from their own annotated data. For individual annotated results, we developed a consistency checking tool. This tool first collects all the annotated phrases and their statistics in the treebank, and it then checks in all of the material for annotation consistency. That is, for any word string forming a phrase, the tool checks the whole treebank to see whether the same word string appearing in different places is bracketed and labeled in the same way. Differences that are detected are verified manually. This tool was found to be useful for checking frequently-used phrases.

5.2 Current Project Status

The corpus currently contains 2,639 articles and a total of 1,035,058 segmented Chinese words. The annotators have identified a total of 282,119 bracketed phrases, including nested phrases. **Table 1** provides statistics about the annotated phrases with different *SS* labels

(mandatory syntactic labels). The annotators have also annotated 98,779 phrases for semantic information.

Table 1. Statistic for annotated phrases with different SS labels

NP	VP	AP	DP	TP	FP	PP	QP
138,785	81,846	16,688	2,812	5,216	2,431	25,198	9,143

All of the annotated material in duplicates has been evaluated against the Gold Standard. On average, the precision of phrase bracketing reached 99.5% and that of recall, 99%. The accuracy achieved in the syntactic labeling of correctly bracketed phrases was, on average, 99.8%, while that of semantic labeling was 98.5%. It was more difficult to determine the accuracy of individually annotated data, that is, of data that was only annotated by one person. Our approach was to randomly select a sample consisting of 5% of the material individually annotated by each annotator. We then annotated these samples in duplicates to evaluate the accuracy of the original annotations. The evaluation results showed that the precision achieved in the phrase bracketing of individually annotated data was 98.8%, while that of recall was 98.2%. The accuracy of syntactic labeling was 99.5% and that of semantic labeling was 98.0%.

6. Applications of The PolyU Treebank

The fact that the PolyU Treebank provides not only syntactic but also semantic information of phrases means that it can be applied to a variety of NLP applications. Of course, the most obvious candidate is the training and testing of an automatic shallow parser [Lu *et al.* 2003]. Other applications in which it can be used are Chinese collocation extraction and research on the acquisition of temporal expressions.

In 2003, our team developed an effective window-based statistical algorithm for extracting Chinese collocation which the precision rate of extracted bigram collocation reached 61% [Xu 2003]. The extraction results included some pseudo-collocations, that is, word combinations that frequently co-occurred but were in fact irrelevant, like the typical 'doctor-nurse' combination in English [Church and Hanks 1990]. The fact that these pseudo-collocations were statistically significant made it difficult to remove them individually using any statistic-based extraction method. However, given that a Chinese collocation normally occurs only within a phrase or between the headwords of relevant phrases [Zhang and Lin 1992], we were able to use the syntactic information, i.e., the boundaries and headword of phrases, recorded in the PolyU Treebank to refine the searching context window, eliminate some pseudo-collocations, and also retrieve some low-frequency collocations.

The PolyU Treebank is currently being used to acquire temporal expressions. The annotated time phrases (*TP*) and the additional annotation with more finely-tuned

point-of-time (*TP-PO*) and period-of-time (*TP-DU*), are very helpful to acquire and classify temporal expressions.

7. Conclusions and Future Work

This paper has described the design and construction of a manually annotated one-million-word Chinese shallow treebank. This is the first attempt to not only construct a large-scale shallow Treebank for use in practical applications but also provide a treebank for a public use.

The PolyU Treebank has four main advantages:

1. It offers a set of practical, shallow annotation specifications with low ambiguity. These specifications can be used to guide both treebank annotation and the development of an automatic shallow parser.
2. The PolyU Treebank provides useful syntactic information, including the boundaries and syntactic categories of base-phrases, nested phrases, and maximal-phrases. Because it adopts a widely accepted grammar framework and makes use of a widely accepted phrase categories, other researchers can readily use the PolyU Treebank.
3. The PolyU Treebank provides useful semantic information, which is unavailable in other syntactic treebanks.
4. The PolyU Treebank offers a large amount of high-quality data.

Presently, we are developing visualization tools that will support user-friendly keyword searching, context indexing, and annotation case searching. We are also keen to include the annotation of semantic information labels for phrases so as to make the PolyU Treebank more useful in a wider range of research applications. Currently, the PolyU Treebank is being used in research on Chinese collocation extraction, Chinese terminology extraction and summarization, and the acquisition of temporal expressions. In these tasks, the syntactic and semantic knowledge obtained from the PolyU Treebank has been found to improve performance. Finally, we intend to make the PolyU Treebank data available for public access in the hope that the availability of, such a large-scale Chinese shallow Treebank will facilitate NLP research.

Acknowledgement

This project was partially supported by The Hong Kong Polytechnic University (Project Code A-P203) and a CERG Grant (Project code 5087/01E). Special thanks go to Mr. Wei Yan for leading the annotation team at Beijing Language and Culture University and to the anonymous reviewers for their valuable comments, which improves the quality and readability of this paper.

References

Chen, F. Y., P. F. Tsai, K. J. Chen, and C. R. Hunag, "Sinica Treebank," *International Journal of Computational Linguistics and Chinese Language Processing,* 4(2), 1999, pp. 183-204.

Chen K. J., C. R. Huang, F. Y. Chen, C. C. Luo, M. C. Chang, C. J. Chen, and Z. M. Gao, "Sinica Treebank: Design Criteria, Representational Issues and Implementation," *Building and Using Parsed Corpora*, ed. by A. Abeillé, Dordrecht: Kluwer, 2003, pp.231-248.

Church, K., and P. Hanks, "Word Association Norms, Mutual Information, and Lexicography," *Computational Linguistics*, 16(1), 1990, pp. 22-29.

Dalemans W. B. Sabine, and V. Jorn, "Memory-based Shallow Parsing," In *Proceedings of Conference on Computational Natural Language Learning* , 1999, Bergen, pp.53-60.

Leech, G. N., and R. Garside, *Running a Grammar Factory: the Production of Syntactically Analyzed Corpora or "Teebanks",* Johansson and Stenstron, 1996.

Li, B. L., Q. Lu, and Y. Li., "Building a Chinese Shallow Parsed Treebank for Collocation Extraction," In *Proceedings of Conference on Intelligent Text Processing and Computational Linguistics,* 2003, Mexico City, pp. 402-405.

Lu, Q., J. Zhou, and R. F. Xu, "Machine Learning Approaches for Chinese Shallow Parsing," In *Proceedings of IEEE International Conference on Machine Learning and Cybernetics,* 2003, Xi'an, China, pp.2309-2314.

Sun, H. L., "A Content Chunk Parser for Unrestricted Chinese Text, " PhD Thesis, Peking University, 2001.

Marcus, M. B. Santorini, and M. A. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics*, 19(1), 1993, pp. 313-330.

Wallis, S., "Completing Parsed Corpora: from Correction to Evolution," *Building and Using Parsed Corpora*, ed. by A. Abeillé , Dordrecht: Kluwer, 2003, pp.61-71.

Xia, F., M. Palmer, N. W. Xue, M. E. Okurowski, J. Kovarik, F. D. Chiou, S. Z. Huang, T. Kroch and M. Marcus, "Developing Guidelines and Ensuring Consistency for Chinese Text Annotation," In *Proceedings of second International Conference on Language Resources and Evaluation,* 2000, Athens, Greece

Xu, R. F., Q. Lu, and Y. Li, "An Automatic Chinese Collocation Extraction Algorithm based on Lexical Statistics," In *Proceedings of International Conference on Natural Language Processing and Knowledge Engineering,* 2003, Beijing, pp.321-326.

Xue, N. W., F. D. Chiou, and M. Palmer,"Building a Large-Scale Annotated Chinese Corpus," In *Proceedings of 17th International Conference on Computational Linguistics*, 2002, Taipei, Taiwan, pp.336-343

Yu, S. W., X. F. Zhu, H. Wang, and Y. Y. Zhang, *The Grammatical Knowledge- base of Contemporary Chinese: A Complete Specification*. Tsinghua University Press, Beijing, China, 1998.

Yu, S. W., et al. “Guideline of People’s Daily Corpus Annotation,” Technical report, Beijing University, 2001.

Zhang, S. K. and X. G. Lin，*Collocation Dictionary of Modern Chinese Lexical Words*, 1st ed., Business Publisher, Beijing, China, 1992.

Appendix 1. Part-of-Speech Tag Set

ag	形容词语素 adjective morpheme	a	形容词 adjective	ad	副形词 adverb-adjective	an	名形词 adnoun
bg	区别语素 distinguish morpheme	b	区别词 distinguish word	c	连词 conjunction	dg	副语素 adverb morpheme
d	副词 adverb	e	叹词 exclamation	f	方位词 position word	h	前缀 heading element
i	成语 Idiom	j	简略语 abbreviation	k	后缀 tail element	l	惯用语 habitual word
mg	数语素 numeral morpheme	m	数词 numeral	ng	名语素 noun morpheme	n	名词 noun
nr	人名 person's name	ns	地名 toponym	nt	组织名 organization noun	nx	外文 foreign character
nz	专有名词 other proper noun	o	拟声词 onomatopoeia	p	介词 preposition	q	量词 quantifier
rg	代语素 pronoun morpheme	r	代词 pronoun	s	方位词 Location word	tg	时语素 time morpheme
T	时间词 time	u	助词 Auxiliary	vg	动语素 verb morpheme	v	动词 verb
vd	副动词 adverb-verb	vn	动名词 gerund	w	符号 punctuation	yg	语气词素 modal morpheme
y	语气词 modal word	z	状态词 state word				