Computational Linguistics and Chinese Language Processing
Vol. 6, No. 1, February 2001, pp. 1-26

Improving Translation Selection with a New Translation Model Trained by Independent Monolingual Corpora

Ming Zhou[*], Yuan Ding[+1], Changning Huang[*]

Abstract

We propose a novel statistical translation model to improve translation selection of collocation. In the statistical approach that has been popularly applied for translation selection, bilingual corpora are used to train the translation model. However, there exists a formidable bottleneck in acquiring large-scale bilingual corpora, in particular for language pairs involving Chinese. In this paper, we propose a new approach to training the translation model by using unrelated monolingual corpora. First, a Chinese corpus and an English corpus are parsed with dependency parsers, respectively, and two dependency triple databases are generated. Then, the similarity between a Chinese word and an English word can be estimated using the two monolingual dependency triple databases with the help of a simple Chinese-English dictionary. This cross-language word similarity is used to simulate the word translation probability. Finally, the generated translation model is used together with the language model trained with the English dependency database to realize translation of Chinese collocations into English. To demonstrate the effectiveness of this method, we performed various experiments with verb-object collocation translation. The experiments produced very promising results.

Keywords: Translation selection, Statistical machine translation, Chinese-English machine translation, Cross language word similarity

1. Introduction

Selecting the appropriate word translation among several options is a key technology of machine translation. For example, the Chinese verb "订" is translated in different ways in

[*] Microsoft Research, Asia.
[+] Tsinghua University.
[1] Currently is studying at University of Pennsylvania as Ph.D. student. This work was done while visiting Microsoft Research Asia as a visiting student.

terms of objects, as shown in the following:

订 报纸 →subscribe to a newspaper
订 计划 →make a plan
订 旅馆 →book a hotel
订 车票 →reserve a ticket
订 时间 →determine the time

In recent years, there has been increasing interest in applying statistical approaches to various machine translation tasks, from MT system mechanisms to translation knowledge acquisition. For translation selection, most researches applied statistical translation models. In such statistical translation models, to get the word translation probability as well as translation templates, bilingual corpora are needed. However, for quite a few languages, large bilingual corpora rarely exist, while large monolingual corpora are easy to acquire. It will be helpful to alleviate the burden of collecting bilingual corpus if we can use monolingual corpora to estimate the translation model and find alternative to translation selection.

We propose a novel approach to this problem in the Chinese-English machine translation module which is to be used for cross-language information retrieval. Our method is based on the intuition that although the Chinese language and the English language have different definitions of dependency relations, the main dependency relations like subject-verb, verb-object, adjective-noun and adverb-verb tend to have strongly direct correspondence. This assumption can be used to estimate the word translation probability. Our proposed method works as follows. First, a Chinese corpus and an English corpus are parsed, respectively, with a Chinese dependency parser and an English dependency parser, and two dependency triple databases are generated as the result. Second, the word similarity between a Chinese word and an English word are estimated with these two monolingual dependency triple databases with the help of a simple Chinese-English dictionary. This cross-language word similarity is used as the succedaneum of the word translation model. At the same time, the probability of a triple in English can be estimated with the English triple database. Finally, the word translation model, working together with the triple probability, can realize a new translation framework. Our experiments showed that this new translation model achieved promising results in improving translation selection. The unique characteristics of our method include: 1) use of two monolingual corpora to estimate the translation model. 2) use of dependency triples as basis for our method.

The remainder of this paper is organized as follows. In Section 2, we give a detailed description to our new translation model. In section 3, we describe the training process of our new model, focusing on the process of constructing the dependency triple database for English and Chinese. The experiments and evaluation of this new method are reported in Section 4. In

Section 5, some related works are introduced. Finally in Section 6, we draw conclusions and discuss future work.

2. A New Statistical Machine Translation Model

In this section, we will describe the proposed translation model. First, we will report our observations from a sample word-aligned bilingual corpus in order to verify our assumption. After that, we will introduce the method for estimating the cross-language word similarity by means of two monolingual corpora. Finally, we will give a formal description of the new translation model.

2. 1 Dependency Correspondence between Chinese and English

A dependency triple consists of a head, a dependant, and a dependency relation between the head and the dependant. Using a dependency parser, a sentence can be analyzed to obtain a set of dependency triples in the following form:

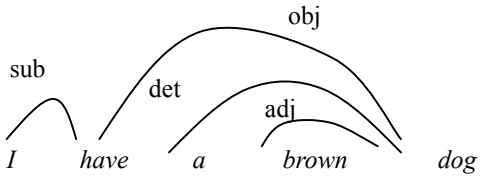
$$trp = (w_1, rel, w_2),$$

which means that word w_1 has a dependency relation of rel with word w_2.

For example, for the English sentence *"I have a brown dog",* a dependency parser obtains a set of triples as follows:

(1)

a.

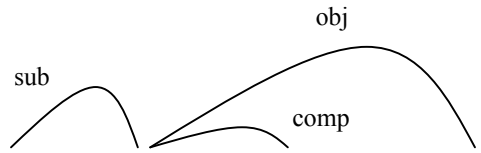


b. (*have, sub, I*), (*I, sub-of, have*), (*have, obj, dog*), (*dog, obj-of, have*), (*dog, adj, brown*), *(brown, adj-of, dog), (dog, det, a), (a, det-of, dog)*[2]

Similarly, for the Chinese sentence "国家颁布了计划", we can get the following dependency triples with a dependency parser:

[2] The standard expression of the dependency parsing result is: (*have, sub, I*), (*have, obj, dog*), (*dog, adj, brown*), *(dog, det, a).*

(2)

a 

.b. (颁布, sub, 国家), (国家, sub-of, 颁布), (颁布, obj, 计划), (计划, obj-of, 颁布), (颁布, comp, 了), (了,comp-of, 颁布)[3]

Among all the dependency relations in Chinese and in English, the key dependency relations are subject-verb (denoted as sub), verb-object (denoted as obj), adjective-noun(denoted as adj) and adverb-verb(denoted as adv). Our intuitive assumption is that although Chinese language and English language have different schemes of dependency relations, these key dependency relations tend to have strong correspondence. For instance, normally, a word pair with subject-verb relation in Chinese can be translated into a subject-verb relation pair in English. Formally speaking, for a triple *(A, D, B)* in Chinese, where *A and B* are words, and *D* is one of the key dependency relations mentioned above, the translation of the triple *(A, D, B*) in English, can be expressed as *(A', D', B')*, where *A'* and *B'* are the translations of *A* and *B,* respectively, and *D'* is the dependency relation between *A'* and *B'* in the English language[4]. Our assumption is that although D and D' may be different in denotation, they can be mapped directly in most cases.

In order to verify our assumption, we conducted an investigation with a Chinese-English bilingual corpus[5]. The bilingual corpus, consisting of 60,000 pairs of Chinese sentences and English sentences selected from newspapers, novels, general bilingual dictionaries and software product manuals, was aligned manually at the word level. An example of the word aligned corpus is given in Table 1. Each word is identified with a number in order to indicate the word alignment information.

[3] The standard expression of the dependency parsing result is: (颁布, sub, 国家), (颁布, obj, 计划), (颁布, comp, 了).

[4] Sometimes to get a better translation, a triple in one language is not translated into a triple in other language, but except in very extreme cases, it will still be acceptable if it is translated into a triple.

[5] This corpus, produced by Microsoft Research Asia, is currently reserved for Microsoft internal use only.

Table 1. *The word aligned bilingual corpus*

Chinese sentence	当/1 斯科特/2 抵达/3 南极/4 的/5 时候/6 ，/7 他/8 发现/9 阿蒙森/10 比/11 他/12 领先/13 。/14
English sentence	When/1 Scott/2 reached/3 the/4 South/5 Pole/6 , /7 he/8 found/9 Amundsen/10 had/11 anticipated/12 him/13 ./14
Aligned word pair	(1,5,6:1); (2:2); (3:3); (4:4,5,6); (7:7); (8:8); (9:9); (10:10); (11:nil); (12:13); (13:12); (14:14);

To obtain statistics of the dependency relation correspondence, we parsed 10,000 sentence pairs with the English parser Minipar [Lin 1993, Lin 1994] and the Chinese parser BlockParser [Zhou 2000]. The parsing results were expressed in dependency triples. We then mapped the dependency relations so that we could count the correspondences between an English dependency relation and a Chinese dependency relation. More than 80% of subject-verb, adjective-noun and adv-verb dependency relations could be mapped, while verb-object correspondence was not so high. We show the verb-object correspondence results in Table 2.

Table 2. *Triple correspondence between Chinese and English.*

Dependency Type	E-C Positive	E-C Negative	Mapping Rate	C-E Positive	C-E Negative	Mapping Rate
Verb-Object	7,832	4,247	64.8%	6,769	3,751	64.3%

"*E-C Positive*" means an English verb-object was translated into a Chinese verb-object. "*E-C Negative*" means an English verb-object was not translated into a Chinese verb-object. The *E-C Positive Rate* reached 64.8% and the *C-E Positive Rate* reached 64.3%. These statistics show that our correspondence assumption is reasonable but not strong. Now we will examine the reasons why some of the dependency relations cannot be mapped directly.

Table 3. *Negative examples of triple mapping.*

Chinese verb-object triple	English translation
够 开销	be enough for
用 数字	in numeral characters
用 货币	Change to currency
名叫 威廉_•_罗	an Englishman, Willian Low
…觉得逃避到生活虽艰苦但比较简朴的年代里是件愉快的事。	…found it pleasant to escape to a time when life, though hard, was relatively simple.

From Table 3, we can see that "negative" mapping has several causes. The most important reasons are: a Chinese verb-object can be translated into a single English verb (e.g., an intransitive verb) or can be translated into verb+prep+obj. If these two mappings (as shown

in Table 4) are also considered reasonable correspondences, then the mapping rate will increase significantly. As seen in Table 5, the *E-C Positive rate* and the *C-E Positive rate* reached 82.71% and 83.87% respectively.

Table 4. *Extended mapping.*

Chinese triple	English triple	Examples
Verb-Object	Verb(usually intransitive verb)	读-书 →read
Verb-Object	Verb+Prep-Object	用-货币→change to – currency

Table 5. *Triple correspondence between Chinese and English.*

Type	E-C Positive	E-C Negative	Mapping rate	C-E Positive	C-E Negative	Mapping Rate
Verb-Object	9991	2088	82．71%	8823	1697	83．87%

This implies that all four key dependency relations can be mapped very well, showing that our assumption is correct. This fact will be used to estimate the word translation model using two monolingual corpora. The method will be given in the following subsections.

2.2 Cross-Language Word Similarity

We will next describe our approach to estimating the word translation likelihood based on the triple correspondence assumption with the help of a simple Chinese-English dictionary. The key idea is to calculate "cross-language similarity", which is an extension of word similarity within one language.

Several statistical approaches to computing word similarity have been proposed. In these approaches, a word is represented by a word co-occurrence vector in which each feature corresponds to one word in the lexicon. The value of a feature specifies the frequency of joint occurrence of the two words in some particular relations and/or in a certain window size in the text. The degree of similarity between a pair of words is computed using a certain similarity (or distance) measure that is applied to the corresponding pairs of vectors. This similarity computation method relies on the assumption that the meanings of the words are related to their co-occurrence patterns with other words in the text. Given this assumption, we can expect that words which have similar co-occurrence patterns will resemble each other in meaning.

Different types of word co-occurrences have been examined with respect to computing word similarity. They can in general be classified into two types, which refer to the co-occurrence of words within the specified syntactic relations, and the co-occurrence of words that have non-grammatical relations in a certain window in the text. The set of

co-occurrences of a word within syntactic relations strongly reflects its semantic properties. Lin [1998b] defined lexical co-occurrences within syntactic relations, such as subject-verb, verb-object, adj-noun, etc. These types of co-occurrences can be used to compute the similarity of two words.

While most methods proposed up to now are for computing the word similarity within one language, we believe that some of these ideas can be extended to computation of "cross-language word similarity". Cross-language word similarity denotes the commonality between one word in a language and one word in another language. In each language, a word is represented by a vector of features in which each feature corresponds to one word in the lexicon. The key to computing cross-language similarity is to determine how to calculate the similarity of two vectors which are represented by words in different languages.

Based on the triple correspondence assumption which we have made in 2.1, dependency triples can be used to compute the cross language similarity. In each language, a word is represented by a vector of dependency triples which co-occur with the word in the sentence. Our approach assumes that a word in one language is similar to a word in another language if their vectors are similar in some sense. In addition, we can use a bilingual lexicon to bridge the words in the two vectors to compute cross-language similarity.

Our similarity measure is an extension of the measure proposed in [Lin, 1998b], where the similarity between two words is defined as the amount of information contained in the commonality between the words and is divided by the sum of information in the descriptions of the two words in each language respectively.

In Lin [1998b]'s work, a dependency parser was used to extract dependency triples. For a word w_1, a triple (w_1, rel, w_2) represents a feature of w_1, which means w_1 can be used in relation of rel with word w_2. The description of a word *w* consists of the frequency counts of all the dependency triples that match the pattern *(w,* , *)*.

An occurrence of a dependency triple (w_1, rel, w_2) can be regarded as the co-occurrence of three events [Lin, 1998b]:

A: a randomly selected word is w_1;

B: a randomly selected dependency type is rel;

C: a randomly selected word is w_2.

According to Lin [1998b], if we assume that A and C are conditionally independent given B, then the information contained in $\| w_1, rel, w_2 \| = f(w_1, rel, w_2) = c$ can be

computed as follows[6]:

$$I(w_1, rel, w_2) = -\log(P_{MLE}(B)P_{MLE}(A \mid B)P_{MLE}(C \mid B)) - (-\log P_{MLE}(A,B,C)); \quad (1)$$

where:

$$P_{MLE}(A \mid B) = \frac{f(w_1, rel, *)}{f(*, rel, *)}; \quad (2)$$

$$P_{MLE}(C \mid B) = \frac{f(*, rel, w_2)}{f(*, rel, *)}; \quad (3)$$

$$P_{MLE}(B) = \frac{f(*, rel, *)}{f(*, *, *)}; \quad (4)$$

$$P_{MLE}(A,B,C) = \frac{f(w_1, rel, w_2)}{f(*, *, *)}; \quad (5)$$

where $f(x)$ denotes the frequency of x; * is a wildcard for all possible combinations.

Finally, we have [Lin, 1998b]

$$I(w_1, rel, w_2) = \log_2 \frac{f(w_1, rel, w_2) f(*, rel, *)}{f(w_1, rel, *) f(*, rel, w_2)} \quad (6)$$

Let $T(w)$ be the set of (rel, w') such that $\log_2 \frac{f(w, rel, w') f(*, rel, *)}{f(w, rel, *) f(*, rel, w')}$ is positive.

Then the similarity between two words, w_1 and w_2, within one language is defined as follows [Lin, 1998b]:

$$Sim(w_1, w_2) = \frac{\sum_{(rel, w) \in T(w_1) \cap T(w_2)} (I(w_1, rel, w) + I(w_2, rel, w))}{\sum_{(rel, w) \in T(w_1)} I(w_1, rel, w) + \sum_{(rel, w) \in T(w_2)} I(w_2, rel, w)} \quad (7)$$

Now, let us see how we can extend to cross language. Similarly, for a Chinese word w_C and an English word w_E, let $T(w_C)$ be the set of pairs (rel_C, w_C') such that $\log_2 \frac{f(w_c, rel_c, w_c') f(*, rel_c, *)}{f(w_c, rel_c, *) f(*, rel_c, w_c')}$ is positive, and let $T(w_E)$ be the set of pairs (rel_E, w_E') such that $\log_2 \frac{f(w_E, rel_E, w_E') f(*, rel_E, *)}{f(w_E, rel_E, *) f(*, rel_E, w_E')}$ is positive. Then we can similarly define cross-language word similarity as follows:

[6] Please see [Lin, 1998b] for the detailed derivation process of this formula.

$$Sim(w_C, w_E) = \frac{I_{common}(w_C, w_E)}{\sum_{(rel_C, w'_C) \in T(w_C)} I(w_C, rel_C, w'_C) + \sum_{(rel_E, w'_E) \in T(w_E)} I(w_E, rel_E, w'_E)} \quad (8)$$

where $I_{common}(w_C, w_E)$ denotes the total information contained in the commonality of the features of w_C and w_E. Actually, we have three different methods for calculating $I_{common}(w_C, w_E)$.

1) Map Chinese into English

We define

$$T_{C \to E}(w_E) = \{(rel_E, w'_E) \mid rel_E = correspondence(rel_C), w'_E \in Tran(w'_C)\} \bigcap T(w_E), where \ \ (rel_C, w'_C) \in T(w_C)$$

$$T_{C \to E}(w_C) = \{(rel_C, w'_C) \mid (rel_E, w'_E) \in T(w_E), where \ rel_E = correspondence(rel_C), w'_E \in Tran(w'_C)\}$$

Here,

$Tran(x)$ denotes the set of possible translations of word x which are defined in the bilingual lexicon and $rel_E = correpondence(rel_C)$ is the English dependency type corresponding to a Chinese dependency type rel_C.

2) Map English into Chinese

Similarly, we define

$$T_{E \to C}(w_C) = \{(rel_C, w'_C) \mid rel_C = correspondence(rel_E), w'_C \in Tran(w'_E)\} \bigcap T(w_C), where \ \ (rel_E, w'_E) \in T(w_E)$$

$$T_{E \to C}(w_E) = \{(rel_E, w'_E) \mid (rel_C, w'_C) \in T(w_C), where \ rel_C = correspondence(rel_E), w'_C \in Tran(w'_E)\}$$

Here,

$rel_C = correpondence(rel_E)$ is the Chinese triple type with rel_C corresponding to an English triple type rel_E.

3) Map both English into Chinese and Chinese into English

Similarly, we define

$$T_{C \leftrightarrow E}(w_C) = T_{E \to C}(w_C) \cup T_{C \to E}(w_C)$$
$$T_{C \leftrightarrow E}(w_E) = T_{E \to C}(w_E) \cup T_{C \to E}(w_E)$$

Then, we can define the cross-language word similarity of w_C and w_E in the following three ways:

$$Sim_{C\to E}(w_C, w_E) = \frac{\sum_{(rel_C, w'_C)\in T_{C\to E}(w_C)} I(w_C, rel_C, w'_C) + \sum_{(rel_E, w'_E)\in T_{C\to E}(w_E)} I(w_E, rel_E, w'_E)}{\sum_{(rel_C, w'_C)\in T(w_C)} I(w_C, rel_C, w'_C) + \sum_{(rel_E, w'_E)\in T(w_E)} I(w_E, rel_E, w'_E)} \quad (9)$$

$$Sim_{E\to C}(w_C, w_E) = \frac{\sum_{(rel_C, w'_C)\in T_{E\to C}(w_C)} I(w_C, rel_C, w'_C) + \sum_{(rel_E, w'_E)\in T_{E\to C}(w_E)} I(w_E, rel_E, w'_E)}{\sum_{(rel_C, w'_C)\in T(w_C)} I(w_C, rel_C, w'_C) + \sum_{(rel_E, w'_E)\in T(w_E)} I(w_E, rel_E, w'_E)} \quad (10)$$

$$Sim_{E\leftrightarrow C}(w_C, w_E) = \frac{\sum_{(rel_C, w'_C)\in T_{E\leftrightarrow C}(w_C)} I(w_C, rel_C, w'_C) + \sum_{(rel_E, w'_E)\in T_{E\leftrightarrow C}(w_E)} I(w_E, rel_E, w'_E)}{\sum_{(rel_C, w'_C)\in T(w_C)} I(w_C, rel_C, w'_C) + \sum_{(rel_E, w'_E)\in T(w_E)} I(w_E, rel_E, w'_E)} \quad (11)$$

Similarity (9) can be seen as the likelihood of translating a Chinese word into an English word, similarity (10) can be seen as the likelihood of translating an English word into a Chinese word, and similarity (11), a balanced and asymmetry formula, can be seen the "neural" similarity of a Chinese word and an English word.

2.3 Translation Selection Model Based on Cross-Language Similarity

We will next discuss how we can build a translation model in order to solve the translation selection problem in dependency triple translation. Suppose we want to translate a Chinese dependency triple $c = (w_{C1}, rel_C, w_{C2})$ into an English dependency triple $e = (w_{E1}, rel_E, w_{E2})$; this is equivalent to finding $e_{\max}$ that will maximize the value $P(e \mid c)$ according to the statistical translation model [Brown, 1993].

Using *Bayes' theorem,* we can write

$$P(e \mid c) = \frac{P(e)P(c \mid e)}{P(c)} \quad (12)$$

Since the denominator $P(c)$ is independent of *e* and is a constant for a given Chinese triple, we have

$$e_{\max} = \arg\max_{e}(P(e)P(c \mid e)) \tag{13}$$

Here, the $P(e)$ factor is a measure of the likelihood of the occurrence of a dependency triple e in the English language. It makes the output of e natural and grammatical. $P(e)$ is usually called the language model, which depends only on the target language. $P(c \mid e)$ is usually called the translation model.

In single triple translation, $P(e)$ can be estimated using formula (5), which can be rewritten as

$$P_{MLE}(w_{E1}, rel_E, w_{E2}) = \frac{f(w_{E1}, rel_E, w_{E2})}{f(*,*,*)}$$

In addition, we have

$$P(c \mid e) = P(w_{C1} \mid rel_C, e) \times P(w_{C2} \mid rel_C, e) \times P(rel_C \mid e)$$

We suppose that the selection of a word in translation is independent of the type of dependency relation, therefore we can assume that w_{C1} is only related to w_{E1}, and that w_{C2} is only related to w_{E2}. Here, we use cross-language word similarity $Sim_{E \to C}$ (see formula 10) to simulate the translation probability from an English word into a Chinese word. Using $Likelihood(c \mid e)$ [7] to replace $P(c \mid e)$, we define

$$Likelihood(c \mid e) = Sim_{E \to C}(w_{C1}, w_{E1}) \times Sim_{E \to C}(w_{C2}, w_{E2}) \times P(rel_C \mid e) \tag{14}$$

$P(rel_C \mid e)$ is a parameter which mostly depends on specific word. But this can be simplified as

$$P(rel_C \mid e) = P(rel_C \mid rel_E)$$

Then we have

$$Likelihood(c \mid e) = Sim_{E \to C}(w_{C1}, w_{E1}) \times Sim_{E \to C}(w_{C2}, w_{E2}) \times P(rel_C \mid rel_E)$$

According to our assumption of correspondence between Chinese dependency relations and English dependency relations, we have $P(rel_C \mid rel_E) \approx 1$. Then we have

---

[7] Since $Likelihood$ is not normalized in [0,1], we do not call it probability to avoid confusion.

$$Likelihood(c \mid e) = Sim_{E \to C}(w_{C1}, w_{E1}) \times Sim_{E \to C}(w_{C2}, w_{E2})$$

Therefore, we have

$$\begin{aligned} e_{\max} &= \arg\max_{e}(P(e) \times P(c \mid e)) \\ &= \arg\max_{e}(P(e) \times Likelyhood(c \mid e)) \\ &= \arg\max_{w_{E1}, w_{E2}}(P(e) \times Sim_{E \to C}(w_{C1}, w_{E1}) \times Sim_{E \to C}(w_{C2}, w_{E2})) \end{aligned} \tag{15}$$

In this formula, we use the English dependency triple sets to estimate $P(e)$, and use the English dependency sets and Chinese dependency sets which are independent of each other, to estimate the translation model based on our dependency correspondence assumption. In the whole process, no manually aligned or tagged corpus is needed.

3. Model Training

To estimate the cross-language similarity and the target language triple probability, both Chinese and English dependency triple sets are required to build. Similar to [Lin 1998b], we also use parsers to extract dependency triples from the text corpus. The workflow of constructing the dependency triple databases is depicted in Fig 1.

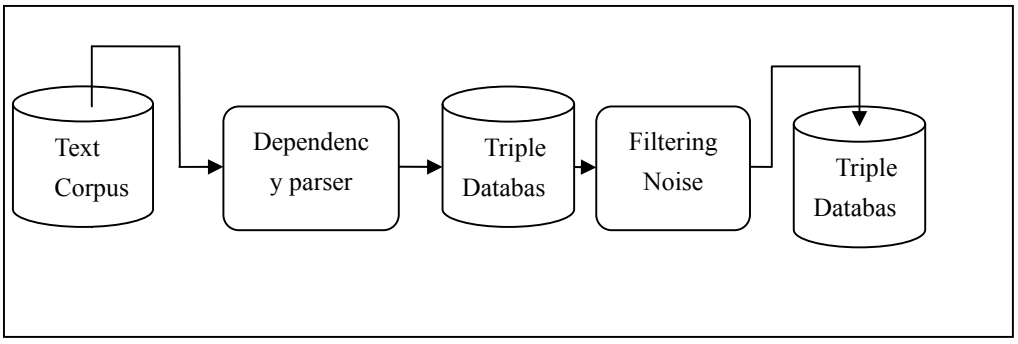


Figure 1 *The flowchart of constructing the dependency triple database.*

As shown in Fig. 1, each sentence from the text corpus is parsed by a dependency parser, and a set of dependency triples is generated. Each triple is put into the triple database. If an instantiation of a type of triple already exists in the triple database, then the frequency of this triple will increase one time. After all the sentences are parsed, we can get a triple database with a large number of triples. Since the parser can not be expected to be 100% correct, some parsing mistakes will inevitably be introduced into the triple database. It is necessary to remove the noisy triples as Lin did [1998a], but in our experiment, we did not apply any noise

filtering technique.

Our English text corpus consists of 750 M (byte) of text from the Wall' Street Journal(1980-1990), and our Chinese text corpus contains 1,200 M(byte) of text from People's Daily (1980-1998). The English parser we used was Minipar [Lin 1993, Lin 1994]. Minipar is a broad-coverage, principle-based parser with a lexicon of more than 90,000 words. The Chinese parser we used here was BlockParser [Zhou 2000]. This is a robust rule parser that breaks up Chinese sentences into *"blocks"*, which are represented by *headwords*. Then syntactical dependency analysis was applied to the *"blocks"*. 17 POS tags and 19 grammatical relations were recognized by this parser, and 220,000 entries were registered in the parsing lexicon.

The 750M (byte) English newspaper corpus was parsed within 50 hours on a machine with 4 Pentium™ III 800 CPU, and the 1200 M (byte) Chinese newspaper corpus was parsed in 110 hours on the same machine. We extracted the dependency triples from the parsed corpus. There were 19 million occurrences of dependency triple in the English parsed corpus, and 33 million occurrences of dependency triples in the Chinese parsed corpus. As a result, we acquired two databases of dependency triples of the two languages. These two databases served as the information source for the translation model training and triple probability, which we have described in the above sections.

Table 6. *shows a summary of the corpora and parsers in Chinese and English.*

Language	**Description**	**Size(bytes)**	**#Triple**	**Parser**
Chinese	People's Daily 1980~1998	1,200M	33,000,000	Block Parser
English	Wall's Street Journal 1980-1990	750M	19,000,000	Minipar

The E-C and C-E dictionaries used here are the bilingual lexicon used in machine translation systems developed by Harbin Institute of Technology[8]. The E-C lexicon contains 78,197 entries, and C-E dictionary contains 74,299 entries.

Since in this paper, we are primarily interested in the selection of translations of verbs, we utilized only three types of dependency relations for similarity estimation, i.e., verb-object, verb-adverb and subject-verb. The symmetric triples "object-of", "adverb-of" and "subject-of" were also used in calculating the translation model and the triple probability. Table 7 shows the statistics of occurrences of the three kinds of dependency relations.

[8] These two lexicons are not publicly available.

Table 7. *Statistics of the three main triples*

Language	Verb-Object	Verb-Adverb	Subject-Verb
Chinese	14,327,358	10,783,139	8,729,639
English	6,438,398	3,011,767	5,282,866

Therefore, a word w is represented by a co-occurrence vector $\{(rel, w_1^{'}, \#), (rel, w_2^{'}, \#)...\}$, where $rel \in \{verb-object, verb-adverb, subj-verb\}$ [9] in which each feature $(rel, w_1^{'}, \#)$ consists of the dependency relation rel, another word $w_1^{'}$ that constructs the dependency relation, and the frequency count #. Then we extracted the word lists from the Chinese triple sets and the English triple sets, and calculated the similarity of each Chinese word and each English word. For similarity, we only calculated the similarity between verbs and between nouns of the two languages. As a result, a large table was constructed recording the cross-language similarity as shown in table 8. S (i,j) is the similarity between a Chinese word C_i and an English word E_j. Please note that we only apply similarity formula (10) since we were interested in the translation likelihood from an English word to a Chinese word, as explained in the previous section.

Table 8. *Cross-language word similarity matrix*

	E_1	E_2	…	E_m
C_1	S_{11}	S_{12}	…	S_{1m}
C_2	S_{21}	S_{22}	…	S_{2m}
…	…	…	…	…
C_n	S_{n1}	S_{n2}	…	S_{nm}

4. Translation Experiments

Please note that in this paper, we only focus on the verb-object triple translation experiments to demonstrate how to improve translation selection. We conducted a set of experiments with several translation models on the verb-object translation. As the baseline experiment, Model A selected the translation of a verb and its object with the highest frequency as the translation output. Model B utilized the target language triple probability but did not apply the translation model. Model C utilized both the target language triple probability and the translation model.

The verb-object translation answer sets were built manually by English experts from the Department of Foreign Languages of Beijing University. For a certain triple, all the plausible translations are given in building the translation evaluation set. Samples of the evaluation sets are shown in Table 9.

[9] We didn't use the dependency relation of adj-noun.

Table 9. *Evaluation sets prepared by human translators*

Verb	Noun	Translation
说	事	talk business
用	手	use hand
看	电影	see film, see movie
看	电视	watch TV
作	贡献	make contribution

The performance was evaluated based on precision, which is defined as

$$precision = \frac{\#correct\ translaion}{\#total\ verb-obj\ triples} \times 100\%$$

4.1 Various Translation Models

Suppose we want to translate the Chinese dependency triple $c = (w_{C1}, rel_C, w_{C2})$ into the English dependency triple $e = (w_{E1}, rel_E, w_{E2})$; this is equivalent to finding $e_{\max}$ that would maximize translation model we have proposed. To test our method, we conducted a series of translation experiments with incrementally enhanced resources. All the translation experiments reported in this paper were conducted with Chinese-English verb-object triple translation.

Model A (selecting the highest-frequency translation)

As the baseline for our experiment, *Model A* simply selected the translation word in the bilingual lexicon which had the highest frequency in the English corpus. It translated *verb* and *object* separately. *Model A* did not utilize the triple probability or the translation model. Formally, Model A can be expressed as

$$e_{\max} = (\underset{w_{e1} e \in Trans(w_{c1})}{\arg\max}\ (freq(w_{E1})), verb-object, \underset{W_{e2} \in Trans(w_{C2})}{\arg\max}\ (freq(w_{E2}))$$

Model B (selecting the translation with the maximal triple probability)

Model B only used the triple probability in target language, neglecting the translation model. It selected the translation of the triple which was most likely to occur in the target language. We have

$$e_{\max} = \arg\max_{e} P(e) = \underset{\substack{w_{E1} \in Trans(w_{Cq}), \\ w_{E2} \in Trans(w_{C2})}}{\arg\max}\ P(w_{E1}, verb-obj, w_{E2})$$

Model C (selecting the translation which fits both the triple probability and the translation model best)

In *Model C*, both the translation model and triple probability were considered. We have

$$e_{\max} = \arg\max_{e} P(e) \times Likelyhood(c \mid e)$$

$$= \arg\max_{\substack{w_{E1} \in Tran(w_{C1}) \\ w_{E2} \in Tran(w_{C2})}} P(w_{E1}, verb-obj, w_{E2}) \times sim_{E \to C}(w_{C1}, w_{E1}) \times sim_{E \to C}(w_{C2}, w_{E2})$$

4.2 Evaluation

We designed a series of evaluations to test the above models. In this subsection, the evaluation results will be reported. To achieve an objective evaluation, we designed three kinds of testing set, 1) high frequency verb and its object, 2) a low frequency verb and its object, and 3) a low frequency verb-object triple. Please note that each selected verb should take a simple noun as its object, the verbs like "是"(be)，"使"(make), "请"(invite), "认为" were not used since their translations were not directly relied on their objects.

Case-I: High-frequency verbs with their objects

We wanted to observe the performance of these models in the translation of verb-objects in which the verbs were high frequency ones. We randomly selected 53 high-frequency verbs (see Appendix I), and randomly extracted certain number of triples of verb-object relation from the Chinese triple database. Totally 730 triples are extracted. The translation results obtained using the various models are shown in Table 10.

Table 10. *Evaluation on verbs of high frequency*

Model	#Correct	Percentage
Model A	393	53.8%
Model B	512	70.1%
Model C	519	71.1%

From these results, we can see that Model B and Model C achieved considerably better translation precision than did Model A. Model C worked a little better than Model B.

Case-II: Translation of low-frequency verbs with their objects

We tested the translation of the triples composed of low-frequency verbs and a noun. We randomly selected 23 low frequency verbs (see Appendix II) and randomly extracted 108 verb-object triples containing these words from the Chinese triple database. The translation results obtained using the various models are shown in Table 11.

Table 11. *Evaluation of verbs of low frequency*

Model	#Correct	Percentage
Model A	61	56.5%
Model B	85	78.7%
Model C	88	81.5%

Case III: Translation of low-frequency triples

We also tested the translation of low-frequency triples. First we selected the following objects: “国家, 同志, 企业, 政府, 记者, 会议, 经济, 群众, 农民, 市场, 政策, 公司, 家, 条件, 地区, 基础, 书, 时间, 项目, 人员, 利益”. Then we selected triples which contained the above words and occurred less than 5 times. Since the set of such low-frequency triples was very large, we randomly selected 340 triples as the evaluation sets. The results are shown in Table 12.

Table 12. *Evaluation of triples of low frequency*

Model	#Correct	Percentage
Model A	182	53.5%
Model B	283	83.2%
Model C	289	85.0%

We can see that our methods obtained very promising results in all the cases.

4.3 Accommodating Lexical Gaps (OOV)

One of the reasons for translation mistakes is the OOV problem, *i.e.*, the best translation is out of vocabulary. Therefore, the translation quality is seriously affected. For example, “展开” has two translations in the translation lexicon: “unfold” and “develop”. However, the triple “展开, verb-object, 进攻”, which should be translated as “launch, verb-object, attack”, cannot be properly produced with the translations given by the dictionary. To solve this problem, we used new methods to get a number of possible translations based on the translations defined in the dictionary and obtained very interesting results.

Model D (Translation expansion using a bilingual lexicon)

For the Chinese verb-object triple $c = (w_{C1}, verb-object, w_{C2})$, we can expand new translations by employing an E-C lexicon and the C-E lexicon circles:

$$Tranl(x) = \{x''' \mid x''' \in Tran(x''), x'' \in Tran(x'), x' \in Tran(x)\} \bigcup Tran(x)$$

Let x be a Chinese words, let x' be the English translation of x defined in the C-E lexicon, let x'' be the Chinese translation of x' defined in E-C lexicon, and let x''' be the English translation of x'' defined in C-E lexicon. Taking “说” as an example, “talk” is one translation based on the C-E lexicon. Then looking up in the E-C lexicon, “说话” is one

translation of "talk". Looking up in the C-E dictionary again, "speak" is one translation of "说话". In this way, "说" is translated as "speak" in addition to the original translation "talk". Model D can be described formally as follows:

$$e_{\max} = \arg\max_{e} P(e) \times Likelyhood(c \mid e)$$
$$= \arg\max_{\substack{w_{E1} \in Tran1(w_{C1}) \\ w_{E2} \in Tran1(w_{C2})}} P(w_{E1}, verb-obj, w_{E2}) \times sim_{E \to C}(w_{C1}, w_{E1}) \times sim_{E \to C}(w_{C2}, w_{E2})$$

Model E (Translation expansion using dependency triple database)

For a Chinese verb-object triple $c = (w_{C1}, verb-object, w_{C2})$, we assume that the translation of object w_{C2} is expanded by *Model D*, *i.e.*,

$$Tran1(w_{C2}) = \{x''' \mid x''' \in Tran(x''), x'' \in Tran(x'), x' \in Tran(w_{C2})\} \bigcup Tran(w_{C2})$$

However, we expand the verb w_{C1} translation in a new way as shown below:

$$Tran2(w_{C1}) = \{w_{E1} \mid I(w_{E1,} verb-object, w_{E2}) \succ 0, where\ w_{E2} = Tran1(w_{C2})\} \cup Tran(w_{C1})$$

To reduce the bad impact of the blind translation expansion of Model E, we try to assign lower probability to the verbs that are expanded out of the bilingual lexicon. We use the following method: the translations given by the bilingual lexicon share a probability of 0.6 and the other possible translations that are expanded using Model E share a probability of 0.4. Suppose P^* is the additionally assigned probability, and suppose there are m translations given by the bilingual lexicon and n translations expanded by model E. We have the following:

$P^* = \frac{0.6}{m}$	If the translation is obtained from the C-E lexicon
$P^* = \frac{0.4}{n}$	If the translation is obtained through expansion of Model E

Then *Model E* can be described as:

$$e_{\max} = \arg\max_{e} P(e) \times Likelyhood(c \mid e)$$
$$= \arg\max_{\substack{w_{E1} \in Tran2(w_{C1}) \\ w_{E2} \in Tran1(w_{C2})}} P(w_{E1}, verb-obj, w_{E2}) \times sim_{E \to C}(w_{C1}, w_{E1}) \times P^* \times sim_{E \to C}(w_{C2}, w_{E2}) \times P^*$$

The evaluation results obtained using Case-I testing set are shown in Table 13. We can find that both Model D and Model E improved the translation precision. Model E is more powerful than Model D.

Table 13. *Evaluation on verbs of high frequency*

Model	**#Correct**	**Percentage**
Model D	526	71.8%
Model E	587	80.1%

Using Model C, "展开进攻" could not be translated correctly, while Model E correctly gave the answer "launch attack". In table 14 and Appendix III, there are more examples showing the cases in which Model E correctly selected translations. (The English translations marked with * are cases where the translations could not be found in the translation lexicon but were generated with Model E only.)

Table 14. *The translation result overcoming OOV*

展开进攻	launch* attack	打主意	make plan
采取行动	Take action	打基础	make foundation
采取办法	adopt* method	打球	play ball
看电视	watch television	打洞	make hole
看书	Read book	打折扣	offer* discount
看节目	See program	打锣	strike gong
打电报	send telegram	博取同情	evoke* sympathy

We also found that the translation performance was influenced by data sparseness of the triple database. Typically, when an English counterpart for a verb-object triple in Chinese could not be found, Model E will yielded 0 for $P(w_{E1}, verb-object, w_{E2})$. For example, "eat twisted crullers", which corresponds to "吃油条" did not appeared anywhere in the English triple set. This will generate very big influence. We shall tackle this problem in the future.

5. Related Works

The key to improving translation selection is to incorporate human translation knowledge into a computer system. One way is for translation experts to handcraft the translation selection knowledge in the form of selection rules and lexicon features. However, this method is time-consuming and cannot ensure high quality in a consistent way. Current commercial MT systems mainly rely on this method. Another way is to let the computer learn the translation selection knowledge automatically by using a large parallel text. A good survey on this research is that of McKeown & Radev [2000]. Some of the contents are quoted here in a condensed way. Smadja *et al.* [1996] created a system called Champolion, which is based on Smadja's collocation extractor, Xtract. Champollion uses a statistical method to translate both flexible and rigid collocations between English and French using the Canadian Hansard corpus. Champollion's output is a bilingual list of collocations ready for use in a machine translation system. Smadja et al. indicated that 78% of the French translations of valid English

collocations were judged to be correct based on three evaluations by human experts. Kupiec [1993] described an algorithm for the translation of a specific kind of collocations, namely, noun phrases. An evaluation of his algorithm has shown that 90% of the 100 highest ranking correspondences are correct.

Selecting the right word translation is related to word sense disambiguation. Most of the research has reported on using supervised methods, which use sense-tagged corpora. Mooney [1996] gave a good quantitative comparison of various methods. Yarowsky [1995] reported an impressive unsupervised-learning result that trains decision lists for binary sense disambiguation. Schutze [1998] also proposed an unsupervised method, which in essence clusters usages of a word. However, although both Yarowsky and Schutze minimized the amount of supervision, their reported results only for very few examples.

Another related field is computer assisted bilingual lexicon (term) construction. A tool for semi-automatic translation of collocations, Termight, wa described by Dagan and Church [1994]. It can be used to aid translators in finding technical term correspondences in bilingual corpora. The method proposed by Dagan and Church uses extraction of noun phrases in English and word alignment to align the head and tail words of noun phrases with words in the other language. A word sequence of words corresponding to the head and tail is produced as the translation. Because it does not rely on statistical correlation metrics to identify the words of the translation, this method allows the identification of infrequent terms that would otherwise be missed owing to their low statistical significance. Fung [1995] used a pattern-matching algorithm to compile a lexicon of nouns and noun phrases between English and Chinese. Wu and Xia [1994] computed a bilingual Chinese-English lexicon. They used the EM algorithm to produce word alignment across parallel corpora and then applied various linguistic filtering techniques to improve the results.

Since large aligned bilingual corpora are hard to acquire due to copyright restrictions and construction expenses, some researchers have proposed methods which do not rely on parallel corpora. Tanaka and Iwasaki [1996] demonstrated how to use nonparallel corpora to choose the best translations among a small set of candidates. Fung [1997] used similarities in the collocates of a given word to find its translation in the other language. Fung [1998] also explored using an IR approach to get translations of new words using non-parallel but comparable corpora. Dagan and Itai [1994] use a second language monolingual corpus for word sense disambiguation. They used a target language model to find the correct word translations.

Most of the methods for statistical machine translation obtain word translation probability by learning from large parallel corpora [Brown *et al.*, 1993]. Very few researchers have tried to use monolingual corpora to train word translation probability. The most similar

work to our approach is that of [Koehn and Knight. 2000]. Using two completely unrelated monolingual corpora and a bilingual lexicon, they constructed a word translation model for 3830 German and 6147 English noun tokens by estimating word translation probabilities using the EM algorithm. In their experiment, they assumed that the word sequence of English and German was the same, so that in the EM iteration step, the language model of the target language could be used. However, their model was only used to test the translation of nouns; they did not conduct experiments on verb translation. They also did not consider syntactic relations. In addition, it is hard to extend their model to other language pair like Chinese and English.

6. Conclusion

We have proposed a new statistical translation model. The unique characteristics of our model are:

1) The translation model is trained using two unrelated monolingual corpora. We have defined the cross- lingual word similarity, which enable us to compute the similarity between a source language word and a target language word with a simple bilingual lexicon, without using bilingual corpora.

2) The translation model is based on dependency triples, not on word level, which is typically used. It can overcome the long distance dependence problem to some extent. Since the translation of a word is often decided based on a syntactic member that may not be adjacent to the word, this method can hopefully improve translation precision compared with the existing word-based model.

3) Based on the new translation model, we have further proposed new models for tackling OOV issue. The experiments showed that Model E, which expands translations using an English triple database, is a promising model for solving the OOV issue. This is very promising too for the application of cross language information retrieval.

Our approach is completely unsupervised, so it is not necessary for the two corpora to be aligned in any way or to be tagged manually with any information. Such monolingual corpora are readily available for most languages, while parallel corpora rarely exist even for common language pairs. So our method can help overcome the bottleneck of acquiring large-scale parallel corpora. Since this method does not rely on specific dependency triples, it can be used to translate other types of triples such as adjective-noun, adverb-verb and verb-complement in the same way. In addition, our method can be used to build a collocation translation lexicon for an automatic translation system.

This triple based translation approach can be further extended to sentence level

translation. Given a sentence, the main dependency triple can be extracted with a parser, and then each triple can be translated using our method. Then, for dependency triples which are specific to the source language, we can apply a rule-based approach. After all the main triples are correctly translated, a target language grammar can be introduced to realize target language generation. This hopefully will enable us to realize sentence skeleton translation system.

There are some interesting topics for future research. First, since we use parsers which inevitably introduce some parsing mistakes into the generated dependency triple databases, we need to find an effective way to filter out mistakes and perform necessary automatic correction. Second, we need to find a more precise translation expansion method to overcome the OOV issue which is caused by the limited coverage of the lexicon. For instance, we can try using translation expansion by employing a thesaurus that is trained automatically with a large corpus or employ a pre-defined thesaurus like WORDNET. Third, triple data sparseness is a big problem; to solve it, we need to apply some approaches used in statistical language models, such as smoothing methods and the class based models.

References

Brown P.F., Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer, "The mathematics of machine translation: parameter Estimation". *Computational Linguistics*, 19(2), 1993, pp. 263-311.

Tanaka K, H Iwasaki, "Extraction of lexical translation from nonaligned corpora." *COLING-96*: *The 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, 1996, pp. 580-585.

Dagan I, K Church, "TERMIGHT: identifying and translating technical terminology". *4th Conference on Applied Natural Language Processing*, Stuttgart, Germany, 1994, pp. 34-40.

Dagan I, Itai, A, "Word sense disambiguation using a second language monolingual corpus", *Computational Linguistics*, 20(4), 1994, pp. 563-596.

Fung P, " A pattern matching method for finding noun and proper noun translations from noisy parallel corpora", *33rd Annual Conference of the Association for Computational Linguistics*, Cambridge, MA, 1995, pp. 236-233.

Fung P, "Using word signature features for terminology translation from large corpora". Ph.D dissertation, Columbia University, 1997, New York.

Fung P and LO Yuen Yee, "An IR approach for translating new words from nonparallel, comparable Texts". *The 36th Annual Conference of the Association for Computational Linguistics*, Montreal, Canada, August 1998, pp. 414—420.

Koehn. P and K. Knight, "Estimating word Translation probabilities from unrelated monolingual corpora using the EM Algorithm", *National Conference on Artificial Intelligence (AAAI)*, 2000, Austin, Texas.

Lin D., "Principle-based parsing without over-generation", *Proceedings of ACL-93*, 1993, pp 112-120, Columbus, Ohio.

Lin D., "Principar-an efficient, broad-coverage, principle-based parser". *Proceedings of COLING-94*, pp. 482-488, Kyoto, Japan, 1994.

Lin D..1998a, "Extracting collocations from test corpora", *First Workshop on Computational Terminology*, Montreal, Canada, 1998.

Lin D., 1998b, "Automatic retrieval and clustering of similar words", *COLING-ACL98,* Montreal, Canada, 1998.

Mckeown, K R, D R Radev, "Collocations", *Handbook of Natural Language Processing*, pp507-523, Edited by Robert Dale, Hermann Moisl, Harold Somers, 2000.

Mooney, R., "Comparative experiments on disambiguation word senses: An illustration of bias in machine learning", *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, *EMNLP, 1996*.

Smadja F., K. R. Mckeown, V Hatzivassiloglou, "Translation collocations for bilingual lexicons: a statistical approach". *Computational Linguistics*, 22:1-38, 1996.

Schutze, H. "Automatic word sense disambiguation rivaling supervised methods", *Computational Linguistics, 24(1):97-123, 1998.*

Wu D., X. Xia, "Learning an English-Chinese lexicon from a parallel corpus", Technology partnerships for Crossing the Language Barrier: *Proceedings of the First Conference of the Association for Machine Translation in the Americas*, Columbia, MD, pp206-213, 1994.

Yarowsky, D. "Unsupervised word sense disambiguation rivaling supervised methods". *In Proceedings of ACL- 33*, pp. 189-196, 1995.

Zhou M., "A Block-Based Robust Dependency Parser for Unrestricted Chinese Text", *2nd workshop on Chinese language processing*, Hong Kong, 2000.

Appendix I High frequency verb list

Frequency	Word	Frequency	Word	Frequency	Word	Frequency	Word
899835	说	380677	来	322078	用	283612	去
211744	看	199602	作	181205	做	175761	想
175658	出	173802	要	129595	占	124164	上
112368	走	111260	问	92357	打	91020	叫
89115	开	84744	吃	83394	下	81221	搞
75946	讲	75753	办	73911	送	68651	找
68639	发	67103	抓	65796	听	64017	买
63468	住	62936	入	61695	拉	61695	订
384590	进行	362678	发展	228207	举行	223702	参加
214557	通过	204081	加强	195157	提出	172647	解决
151354	组织	133191	采取	126557	开展	110076	发挥
103009	达到	99867	完成	91401	介绍	68801	扩大
68588	计划	67446	引起	60426	恢复	60237	减少
60087	制定						

Appendix II Low frequency verb list

Frequency	Word	Frequency	Word	Frequency	Word	Frequency	Word
2108	践踏	2087	施加	2056	逼近	1555	调配
1549	共享	1498	扣押	1420	反驳	1402	高唱
1389	迷惑	1368	窃	460	遨游	458	规劝
457	胁迫	439	修剪	438	抄袭	304	驯服
294	调遣	278	描摹	270	剽窃	262	吸吮
158	赎回	156	暗藏	153	博取		

Appendix III Some translation results obtained with model E

√	打|锣→strike|gong	√	订|约会→order|appointment	√	做|翻译→make|translation
×	打|鼓→have|drummer	√	订|条约→sign|pact	×	做|演员→do|actor
√	打|钟→play|bell	√	订|计划→make|plan	×	做|保姆→get|housekeeper
√	打|铃→play|bell	√	订|措施→order|measure	×	做|教师→give|teacher
√	打|铁→produce|iron	×	订|日期→order|date	×	做|厨房→do|kitchen
√	打|人→beat|person	√	订|指标→order|target	√	做|纸→make|paper
√	打|仗→do|fight	×	订|制度→order|system	√	看|电影→see|film
×	打|架→buy|shelf	√	订|合同→sign|contract	√	看|电视→watch|television
√	打|脸→beat|face	√	订|契约→sign|charter	√	看|京剧→watch|Bejing opera
×	打|手→play|hand	√	订|公约→sign|pact	√	看|展览→see|exhibition
√	打|头→strike|head	√	订|条件→order|condition	√	看|人→see|person

√	打|枪→fire|gun	√	订|同盟→form|alliance	√	看|书→read|book
√	打|炮→use|cannon	×	订|婚→attend|wedding	√	看|报→read|newspaper
√	打|雷→bring|thunder	√	订|书→order|book	√	看|小说→read|novel
√	打|信号→send|signal	√	订|报→order|newspaper	√	看|文件→see|document
√	打|电话→make|telephone	√	订|杂志→order|magazine	√	看|朋友→see|friend
×	打|靶→hit|target	√	订|票→order|ticket	√	看|学生→see|student
×	打|气→strike|air	√	订|机器→order|machine	×	看|眼睛→see|eye
×	打|针→share|needle	√	订|货→order|goods	√	看|问题→see|problem
√	打|鸟→catch|bird	×	订|本子→carry|notebook	√	看|现象→see|phenomenon
√	打|鱼→catch|fish	×	订|报纸→publish|newspaper	√	看|脸色→see|expression
×	打|老虎→buy|tiger	√	作|打算→make|plan	√	看|本质→see|nature
√	打|蜡→strip|wax	√	作|结论→make|conclusion	×	出|大门→put forth|front door
√	打|草稿→make|draft	√	作|报告→write|report	×	出|国→produce|country
√	打|基础→make|foundation	×	作|斗争→have|struggle	√	出|院→leave|yard
√	打|主意→catch|decision	√	作|曲→write|melody	×	出|城→issue|city
×	打|算盘→work out|abacus	√	作|诗→write|poem	√	出|海→go|sea
×	打|伞→buy|umbrella	√	作|文章→write|article	√	出|境→leave|state
×	打|旗子→play|banner	√	做|鞋→make|shoes	×	出|洞→fill|cavity
×	打|灯笼→sell|lantern	√	做|衣服→make|clothes	×	出|厂→include|works
×	打|饭→ work out|cooked rice	√	做|裤子→make|trousers	×	出|站→make|stop
√	打|酒→buy|wine	√	做|活→do|work	×	出|场→issue|place
√	打|酱油→buy|soy	√	做|菜→make|food	×	出|血→produce|blood
√	打|票→buy|ticke	×	做|饭→make|cooked-rice	×	出|轨→build|rail
×	打|醋→prefer|vinegar	√	做|面包→make|bread	√	出|界→exceed|limit
√	打|柴→collect|firewood	√	做|点心→ make|refreshments	√	出|格→exceed|standard
√	打|草→pack|straw	√	做|工→do|work	√	出|范围→exceed|scope
×	打|麦子→buy|wheat	×	做|沙发→sit|sofa	√	出|主意→produce|idea
√	打|粮食→collect|grain	√	做|生意→make|trade	√	出|题目→issue|subject
√	打|牌→play|cards	√	做|买卖→do|business	√	出|证明→produce|proof
×	打|拳→make|fist	√	做|工作→do|work	×	出|力→produce|power
√	打|哈欠→draw|yawn	√	做|试验→do|test	×	出|钱→issue|money
√	打|盹→have|doze	√	做|事情→do|business	√	出|广告→ produce|advertisement
×	打|冷战→work out|cold war	√	做|功课→do|homework	√	出|劳动力→put forth|labour

√	打|官司→fight|lawsuit	√	做|作业→do|homework	√	出|通知→issue|notice
√	打|井→dig|well	√	做|练习→do|exercise	√	出|节目→produce|program
√	打|洞→make|hole	√	做|学生→become|student	√	出|榜→issue|announcement
√	打|包裹→work out|parcle	×	做|老师→give|teacher	√	出|煤→produce|coal
√	打|行李→pack|luggage	×	做|父亲→do|father	√	出|棉花→produce|cotton
×	打|毛衣→ work out|woolen clothes	√	做|主席→become|chairman	√	出|花生→produce|peanut
√	打|比方→use|analogy	×	做|官→ make|government offcial	√	出|英雄→become|hero

*The √ means correct translation or sometimes acceptable translation, while × means wrong translation.

Computational Linguistics and Chinese Language Processing
Vol. 6, No. 1, February 2001, pp. 27-60

The Use of Clustering Techniques for Language Modeling – Application to Asian Language

Jianfeng Gao[*], Joshua T. Goodman[+], Jiangbo Miao[**]

Abstract

Cluster-based *n*-gram modeling is a variant of normal word-based *n*-gram modeling. It attempts to make use of the similarities between words. In this paper, we present an empirical study of clustering techniques for Asian language modeling. Clustering is used to improve the performance (i.e. perplexity) of language models as well as to compress language models. Experimental tests are presented for cluster-based trigram models on a Japanese newspaper corpus and on a Chinese heterogeneous corpus. While the majority of previous research on word clustering has focused on how to get the best clusters, we have concentrated our research on the best way to use the clusters. Experimental results show that some novel techniques we present work much better than previous methods, and achieve more than 40% size reduction at the same level of perplexity.

1. Introduction

Statistical language modelling (SLM) has been successfully applied in many domains, such as speech recognition, optical character recognition, machine translation, spelling correction, information retrieval, and spoken language understanding [Jelinek, 1990; Church, 1988; Brown *et al*., 1990; Kernighan *et al*., 1990; Miller *et al*., 1999; Zue, 1995]. The dominant technology in SLM is *n*-gram models.

Typically, *n*-gram models are trained on very large corpora. In constructing *n*-gram models, we always face two problems. First, for a general domain model, large amounts of training data can lead to models that are too large for realistic applications. On the other hand, for specific domains, *n*-gram models usually suffer from the data sparseness problem because large amounts of domain-specific data are usually not available.

When *n*-gram models are used, we can define clusters for similar words in a corpus. We

* Microsoft Research, Asia, Beijing, 100080, P.R.C. E-mail: jfgao@microsoft.com
+ Microsoft Research, Redmond Washington 98052, USA. E-mail: joshuago@microsoft.com
** Department of Computer & Information Sciences University of Delaware, USA. This work was done while the author was visiting Microsoft Research Asia.

thus extend word-based *n*-gram models to cluster-based *n*-gram models. This has been demonstrated as an effective way to handle the data sparseness problem. Recent research also shows that cluster-based *n*-gram models are effective for rapid domain adaptation, training on small data sets, and reducing the memory requirements for realistic applications.

Extending our previous work in [Goodman, 2001; Gao *et al*., 2001; Goodman and Gao, 2000], this paper presents an empirical study of clustering techniques for Asian language modeling. Clustering is used to improve the performance (i.e. perplexity) of language models as well as to compress language models. Experimental tests will be presented for cluster-based trigram models on a Japanese newspaper corpus of more than 10 million words, and on a Chinese heterogeneous corpus of more than 11 million characters. The majority of the previous research on word clustering has focused on how to get the best clusters. We have concentrated our research on the best way to use the clusters. Experimental results show that some novel techniques work much better than previous methods.

This paper is structured as follows: In the remainder of this section, we present an introduction to *n*-gram models, smoothing, and performance evaluation. In Section 2, we briefly review previous work on word clustering and cluster-based *n*-gram models. In Section 3, we present our technique of using clusters for trigram models. In Section 4, we describe our method for finding clusters. In Section 5, we present the results of our main experiments. Finally, we present our conclusions in Section 6.

1.1 *N*-gram models

The classic task of *language modeling* is to predict the next word given the previous words. The *n*-gram model is the usual approach. It states the task of predicting the next word as attempting to estimate the conditional probability:

$$P(w_n) = P(w_n \mid w_1 \cdots w_{n-1}) . \tag{1}$$

In practice, the cases of *n*-gram models that people usually use are *n*=2,3,4, referred to as a *bigram*, a *trigram*, and a *four-gram* model, respectively. For example, in trigram models, the probability of a word is assumed to depend only on the two previous words:

$$P(w_n \mid w_1 \cdots w_{n-1}) \approx P(w_n \mid w_{n-2} w_{n-1}) . \tag{2}$$

An estimate of the probability $P(w_i \mid w_{i-2} w_{i-1})$ is given by Equation (3), called the *maximum likelihood estimation* (MLE):

$$P(w_i \mid w_{i-2} w_{i-1}) = \frac{C(w_{i-2} w_{i-1} w_i)}{C(w_{i-2} w_{i-1})} , \tag{3}$$

where $C(w_{i-2} w_{i-1} w_i)$ represents the number of times the sequence $w_{i-2} w_{i-1} w_i$ occurs in the

training text.

A difficulty with this approximation is that for word sequences that do not occur in the training text, where $C(w_{i-2}w_{i-1}w_i) = 0$, the predicted probability is 0. This makes it impossible for a system, such as a speech recognition system, to accept such a 0 probability sequence. Thus, these probabilities are typically smoothed [Chen and Goodman, 1999]: some probability is removed from all non-zero counts, and used to add probability to the 0 count items. The added probability is typically in proportion to some less specific, but less noisy model. For trigram models, typically, a formula of the following form is used:

$$P(w_i \mid w_{i-2}w_{i-1}) = \begin{cases} \dfrac{C(w_{i-2}w_{i-1}w_i) - D(C(w_{i-2}w_{i-1}w_i))}{C(w_{i-2}w_{i-1})} & \text{if } C(w_{i-2}w_{i-1}w_i) > 0 \\ \alpha(w_{i-2}w_{i-1})P(w_i \mid w_{i-1}) & \text{otherwise} \end{cases}, \quad (4)$$

where $\alpha(w_{i-2}w_{i-1})$ is a normalization factor, and is defined in such a way that the probabilities sum to 1. The function $D(C(w_{i-2}w_{i-1}w_i))$ is a discount function. It can, for instance, have a constant value, in which case the technique is called "Absolute Discounting", or it can be a function estimated using the Good-Turing method, in which case the technique is called Good-Turing or Katz smoothing [Katz, 1987; Chen and Goodman 1999].

1.2 Performance evaluation

The most common metric for evaluating a language model is *perplexity*. Formally, the word perplexity PP_W of a model is the reciprocal of the geometric average probability assigned by the model to each word in the test set. It is defined as

$$PP_W = 2^{-\frac{1}{N_W}\sum_{i=1}^{N_W}\log_2 P(w_i|w_{i-2}w_{i-1})}, \quad (5)$$

where N_W is the total number of words in the test set. The perplexity can be roughly interpreted as the geometric mean of the branching factor of the test document when presented to the language model. Clearly, lower perplexities are better.

For applications, such as speech recognition, handwriting recognition, and spelling correction, it is generally assumed that lower perplexity correlates with better performance. In [Gao *et al*., 2001], we presented results that indicate this correlation is especially strong when the *n*-gram model is applied to the application of pinyin to Chinese character conversion, which is a similar problem to speech recognition.

2. Word Cluster and Cluster-based N-grams

For any given assignment of a word w_i to a cluster (also called a class) c_i, there may be many to many mappings; i.e. a word w_i may belong to more than one cluster, and a cluster c_i will

typically contain more than one word. For the sake of simplicity, in this paper, we assume that a word w_i can only be uniquely mapped to its own cluster c_i, which is called hard clustering. The cluster-based *n*-gram model is a variant of the word-based *n*-gram model that uses the frequency of sequences of clusters to help produce a more knowledgeable estimate of the probability of word strings. The basic cluster-based *n*-gram model defines the conditional probability of a word w_i based on its history as the product of the two factors: the probability of the cluster given the preceding clusters, and the probability of a particular word given the cluster [Brown *et al*., 1990]. For example, in cluster-based trigram models, we have

$$P(w_i \mid w_{i-2}w_{i-1}) = P(w_i \mid c_i) \times P(c_i \mid c_{i-2}c_{i-1}) . \tag{6}$$

The MLE of the probability of the word given the cluster, and the probability of the cluster given the two previous clusters can be computed as follows:

$$P(w_i \mid c_i) = \frac{C(w_i)}{C(c_i)}, \tag{7}$$

$$P(c_i \mid c_{i-2}c_{i-1}) = \frac{C(c_{i-2}c_{i-1}c_i)}{C(c_{i-2}c_{i-1})} . \tag{8}$$

A large amount of previous research has focused on how to best cluster similar words together. The proposed methods can be roughly grouped into two categories: (1) knowledge based clustering, and (2) data-driven clustering.

In knowledge based clustering, words are clustered based on the syntactic/semantic information we have for the language and the task [Jelinek, 1990; Heeman, 1999; Heeman and Allen, 1997; Placeway *et al*., 1993; Issar and Ward, 1994; Ward and Young, 1993]. For example, part of speech (POS) tags can be generally used to produce a small number of clusters although this may lead to significantly increased perplexity [Srinivas, 1996; Niesler *et al*., 1998]. Alternatively, if we have domain knowledge, it is often advantageous to cluster words that have a similar semantic functional role together. For example, [Issar and Ward, 1994] used tags like CITY and AIRLINE for an airline information system. There has also been some interesting research on word clustering for Chinese language. For example, [Yang *et al*., 1994] present a method in which Chinese words are simply clustered according to their starting and ending characters. It assumes that because almost every Chinese character is a morpheme with its own meaning, very often words having the same starting or ending characters share some common linguistic properties and, thus, can form a word cluster. A good example is the cluster containing “yesterday” (昨天), “tomorrow” (明天), “everyday” (每天), “Sunday” (星期天) etc.

In data-driven clustering, words are clustered automatically in a such way that the overall perplexity of the corpus is minimized [Brown *et al*., 1992]. A greedy search algorithm

is generally used for clustering. It basically works as follows. First, each word is initialized to a random cluster. Then, at each iteration, every word is moved to a cluster such that the resulting model has the minimum perplexity. The algorithm converges when no single word can be moved to another cluster in a way that reduces the perplexity of the cluster-based *n*-gram model. Most previous research has found only small differences between different techniques for finding clusters [Kneser and Ney, 1993; Yamamoto and Sagisaka, 1999; Ueberla, 1996; Pereira *et al*., 1993; Bellegarda *et al*., 1996; Bai *et al*., 1998]. One result, however, is that automatically derived clusters outperform POS tags [Niesler *et al*., 1998], at least when there is enough training data [Ney *et al*., 1994].

While cluster-based *n*-gram models often offer no perplexity reduction in comparison to word-based *n*-gram models, it is beneficial to smooth the word-based *n*-gram model via either backoff or interpolation methods (although the improvement is marginal) [Maltese and Mancini, 1992; Miller and Alleva, 1997]. One typical example is a combined model where the cluster-based *n*-gram model can be linearly interpolated with a normal word-based *n*-gram model [Brown *et al*., 1992]:

$$\lambda P(w_i \mid w_{i-2}w_{i-1}) + (1-\lambda)P(w_i \mid c_i) \times P(c_i \mid c_{i-2}c_{i-1}) \tag{9}$$

where λ is the interpolation weight optimized on heldout data.

In this study, we focused our research on novel techniques for using clusters rather than different ways of finding clusters. We also noticed that all realistic applications have memory constraints. Therefore, we concentrated our experiments on finding the best way to use cluster-based *n*-gram models together with word-based *n*-gram models to seek the optimum balance between memory storage and perplexity. In Section 5, most of our experimental results will be presented in the form of size/perplexity curves.

3. Using Clusters

In this section, we will describe our techniques for using clusters, which are a bit different than traditional clustering as shown in Equation (6). As a typical example, consider the trigram probability $P(w_3|w_1w_2)$, where w_3 is the word to be predicted, called the *predicted word*, and w_1 and w_2 are context words used to predict w_3, called the *conditional word*. Either the predicted word or the conditional word can be clustered when building cluster-based trigram models. Therefore, there are three basic forms of cluster-based trigram models. When using clusters for the predicted word as shown in Equation (10), we get the first kind of cluster-based trigram model, called *predictive clustering*. When using clusters for the conditional word as shown in Equation (11), we get the second model, called *conditional clustering*. When using clusters for both the predicted word and the conditional word, we get Equation (12), called *combined clustering*:

$$P(w_i \mid w_{i-2}w_{i-1}) = P(c_i \mid w_{i-2}w_{i-1}) \times P(w_i \mid w_{i-2}w_{i-1}c_i)\,, \tag{10}$$

$$P(w_i \mid w_{i-2}w_{i-1}) = P(w_i \mid c_{i-2}c_{i-1})\,, \tag{11}$$

$$P(w_i \mid w_{i-2}w_{i-1}) = P(c_i \mid c_{i-2}c_{i-1}) \times P(w_i \mid c_{i-2}c_{i-1}c_i)\,. \tag{12}$$

In what follows, each technique will be discussed in detail, and illustrated by an example.

3.1 Predictive clustering

Consider a probability such as *P*(*Tuesday*| *party on*). Perhaps the training data contains no instances of the phrase "*party on Tuesday*", although other phrases such as "*party on Wednesday*" and "*party on Friday*" do appear. We can put words into clusters, such as the word "*Tuesday*" into the cluster *WEEKDAY*. Now, we can consider the probability of the word "*Tuesday*" given the phrase "*party on*", and also given that the next word is a *WEEKDAY*. We will denote this probability by *P*(*Tuesday* | *party on WEEKDAY*). We can then decompose the probability

$$P(\textit{Tuesday} \mid \textit{party on}) \;=\; P(\textit{WEEKDAY} \mid \textit{party on}) \times \mathrm{P}(\textit{Tuesday} \mid \textit{party on WEEKDAY}).$$

When each word belongs to only one cluster, this decomposition is a strict equality. This can be trivially proven as follows:

$$\begin{aligned} P(c_i \mid w_{i-2}w_{i-1}) \times P(w_i \mid w_{i-2}w_{i-1}c_i) &= \frac{P(w_{i-2}w_{i-1}c_i)}{P(w_{i-2}w_{i-1})} \times \frac{P(w_{i-2}w_{i-1}c_iw_i)}{P(w_{i-2}w_{i-1}c_i)} \\ &= \frac{P(w_{i-2}w_{i-1}c_iw_i)}{P(w_{i-2}w_{i-1})}\,. \end{aligned} \tag{13}$$

Now, since each word belongs to a single cluster, $P(c_i|w_i)$=1, it follows that

$$\begin{aligned} P(w_{i-2}w_{i-1}c_iw_i) &= P(w_{i-2}w_{i-1}w_i) \times P(c_i \mid w_{i-2}w_{i-1}w_i) \\ &= P(w_{i-2}w_{i-1}w_i) \times P(c_i \mid w_i) \\ &= P(w_{i-2}w_{i-1}w_i)\,. \end{aligned} \tag{14}$$

Substituting Equation (14) into Equation (13), we get

$$P(c_i \mid w_{i-2}w_{i-1}) \times P(w_i \mid w_{i-2}w_{i-1}c_i) = \frac{P(w_{i-2}w_{i-1}w_i)}{P(w_{i-2}w_{i-1})} = P(w_i \mid w_{i-2}w_{i-1})\,. \tag{15}$$

Now, although Equation (15) is a strict equality, when smoothing is taken into consideration, using the clustered probability will be more accurate than using the non-clustered probability. For instance, even if we have never seen an example of "*party on*

Tuesday", perhaps we have seen examples of other phrases, such as "*party on Wednesday*"; thus, the probability $P(WEEKDAY \mid party\ on)$ will be relatively high. Furthermore, although we may never have seen an example of "*party on WEEKDAY Tuesday*", after we backoff or interpolate with a lower order model, we may able to accurately estimate $P(Tuesday|on\ WEEKDAY)$. Thus, our smoothed clustered estimate may be a good one. We call this particular kind of clustering *predictive clustering*. The general form is Equation (10).

3.2 Conditional clustering

On the other hand, we can also cluster the words we are conditioning on. For instance, if "*party*" is in the cluster *EVENT* and "*on*" is in the cluster "*PREPOSITION*", then we can write

$$P(Tuesday \mid party\ on) \approx P(Tuesday \mid EVENT\ PREPOSITION).$$

We call this kind of clustering *conditional clustering*. The general form is Equation (11).

3.3 Combined clustering

It is also possible to combine both predictive and conditional clustering, and, in fact, for some applications, this combination works better than either one separately. Thus, we can compute

$$P(Tuesday \mid party\ on) = P(WEEKDAY \mid EVENT\ PREPOSITION) \times P(Tuesday \mid EVENT\ PREPOSITION\ WEEKDAY).$$

We call this kind of clustering *combined clustering*. The general form is Equation (12). Equation (12) is a generalization of predictive clustering of Equation (10), in which case we used no clustering for conditional words. Equation (12) is also a generalization of conditional clustering of Equation (11), in which case we used no clustering for predicted words. Also notice that the combined cluster-based trigram model of Equation (12) is actually a generalization of a technique invented at IBM (Brown *et al*., 1992), which uses the approximation $P(w_i|c_{i-2}\ c_{i-1}\ c_i) \approx P(w_i|c_i)$ to get

$$P(Tuesday \mid party\ on) \approx P(WEEKDAY|\ EVENT\ PREPOSITION) \times P(Tuesday \mid WEEKDAY).$$

The approximation is suboptimal unless we use high (count) cutoffs for bigrams and trigrams. Given that combined clustering uses more information than regular IBM clustering, we assumed that it would lead to improvements. As will be shown in Section 5, it works about the same or a little better, at least when interpolated with a normal word-based trigram model.

4. Finding Clusters

As described in Section 2, a large number of techniques for finding clusters have been proposed, but previous studies showed that no one technique outperforms other significantly. In this study, we did not explore different techniques for finding clusters, but simply picked

one we thought would be good, based on previous research.

There is no need for the clusters used for different positions to be the same. In particular, for a model like IBM clustering, with $P(w_i|c_i)\times P(c_i|c_{i-2}\ c_{i-1})$, we call the cluster c_i a predictive cluster, and the clusters c_{i-2} and c_{i-1} conditional clusters. The predictive and conditional clusters can be different [Yamamoto and Sagisaka, 1999]. For instance, consider a pair of words like "*a*" and "*an*". In general, "*a*" and "*an*" can follow the same words, and so, for predictive clustering, belong to the same cluster. However, there are very few words that can follow both "*a*" and "*an*", and so, for conditional clustering, they belong to different clusters. We have also found in experiments that the optimal numbers of clusters used for predictive and conditional clustering are different. In this paper, we always optimize both the number of conditional and predictive clusters separately, and reoptimize for each technique on each training data set. This is a very time consuming task, since each time the number of clusters is changed, the models must be rebuilt from scratch. We always try numbers of clusters that are powers of 2, e.g. 1, 2, 4, etc. This seems to produce numbers of clusters that are close enough to optimal.

The clusters are found automatically using a tool that attempts to minimize perplexity. In particular, for conditional clusters, we try to minimize the perplexity of the training data for a bigram of the form $P(w_i|c_{i-1})$, which is equivalent to maximizing

$$\prod_{i=1}^{N} P(w_i \mid c_{i-1}) . \tag{16}$$

For predictive clusters, we try to minimize the perplexity of the training data of $P(c_i|w_{i-1})\times P(w_i|c_i)$. We do not minimize $P(c_i|w_{i-1})\times P(w_i|w_{i-1}\ c_i)$ because we are doing our minimization on unsmoothed training data, and the latter formula would, thus, be equal to $P(w_i|w_{i-1})$ for any clustering. If we were to use the method of leaving-one-out (Kneser and Ney, 1993), then we could use the latter formula, but the approach would be more difficult. Now,

$$\begin{aligned}\prod_{i=1}^{N} P(c_i \mid w_{i-1}) \times P(w_i \mid c_i) &= \prod_{i=1}^{N} \frac{P(w_{i-1}c_i)}{P(w_{i-1})} \times \frac{P(c_i w_i)}{P(c_i)} \\ &= \prod_{i=1}^{N} \frac{P(c_i w_i)}{P(w_{i-1})} \times \frac{P(w_{i-1}c_i)}{P(c_i)} \\ &= \prod_{i=1}^{N} \frac{P(w_i)}{P(w_{i-1})} \times P(w_{i-1} \mid c_i) .\end{aligned} \tag{17}$$

Now, $\frac{P(w_i)}{P(w_{i-1})}$ is independent of the clustering used. Therefore, for selection of the best clusters, it is sufficient to try to maximize $\prod_{i=1}^{N} P(w_{i-1} \mid c_i)$. This is very convenient since it is exactly the opposite of what was done for conditional clustering. It means that we can use

the same clustering tool for both, and simply switch the order used by the program used to get the raw counts for clustering. We give more details about the clustering algorithm in Appendix B.

5. Results and Discussion

In this section, we will report our main experiments. In Section 5.1, we will describe the text corpus we used. In Section 5.2, we will compare the performance of word-based trigram models with cluster-based *n*-gram models. We will give perplexity results of cluster-based *n*-gram models alone, as well as of combined models, where the cluster-based *n*-gram models were interpolated with word-based *n*-gram models. In Section 5.3, we will present a fairly thorough comparison of different techniques for using clusters in language model compression. We will then show that our novel clustering techniques can produce much smaller models at a given level of perplexity.

5.1 Corpora

We performed our experiments on both Chinese and Japanese text corpora. In both cases, we built language models on training data sets of medial size. We performed parameter optimization on a separate set of heldout data, and performed testing on a set of test sets. None of the three data sets overlapped. Out-of-vocabulary words were not included in perplexity computations.

For the Chinese corpus, we used the IME corpus for language model training. It is a balanced corpus, and it exhibits great variety in domain as well as in style. It was collected from the Microsoft input method editor (IME – a software layer that converts keystrokes into Chinese character) tasks. It consists of 11 million characters (or 7 million words after word segmentation). We used 10,000 words for heldout data, and 20,000 words for testing data. The heldout and test data set were every 50th sentence from two non-overlapping sets of an independent open test set. The open test set was carefully designed, and contains approximately half a million characters that have been proofread and balanced among domains, styles, and time [Gao *et al*., 2001]. The lexicon we used is defined by Chinese linguists, with 50,180 entries. The experiments on the Chinese corpus were fairly open tests since we used heterogeneous (in terms of domain and style) data sets from different sources for language model training and testing. Thus, we assumed that problems due to data sparseness and training-test mismatch would be relatively serious.

For experiments on Japanese language modeling, we used a subset of the Nikkei newspaper corpus. In particular, we used the most recent ten million words of the Nikkei corpus for training. As in the Chinese case, we used 10,000 words for heldout data, and 20,000 words for testing data. The heldout and test data sets were every 50th sentence from

two non-overlapping sets, taken from another section of the Nikkei corpus. The lexicon we used contains 180,187 Japanese words. The experiments on the Japanese corpus were more like closed tests since we used homogeneous (at least in terms of style) data sets from the same corpus for language model training and testing. We then assumed that data sparseness and training-test mismatch would be less serious than they would be for the Chinese corpus. We also assumed that the Japanese lexicon was far more complete than the Chinese one. A certain number of the entries in the Japanese lexicon are expressions (e.g. of time and date).

Using the abovementioned Chinese and Japanese text corpora, we sought to test the robustness of our clustering techniques for different languages, corpora, and word sets (e.g. lexicons).

5.2 Clustering for language model improvement

The techniques for finding clusters described in Section 4 were applied to the training corpus to determine suitable word clusters. The word clusters obtained were used to define a cluster-based trigram model and to compute the perplexity on the test sets.

In the experiments, the clustering technique we used created a binary branching tree with words at the leaves. By cutting the tree at a certain level, it was possible to achieve a wide variety of different numbers of clusters. For instance, if the tree was cut after the 8^{th} level, there would be roughly 2^8=256 clusters. Since the tree would not be balanced, the actual number of clusters could be somewhat smaller. Therefore, in what follows, we will use the level of the tree to represent approximately the number of clusters, such as 2^1, 2^2, 2^3, etc. Many more details about the clustering techniques used are given in Appendix B.

5.2.1 Using cluster-based trigram models alone

In the first series of experiments, we used the traditional cluster-based trigram model of Equation (6) to compute the perplexity. The results are shown in Table 1 for the Chinese and the Japanese corpora. For the sake of comparison, the perplexities of the word trigram models are included. In addition, the perplexities of several human defined word clusters sets are shown as well. These include (1) the 28 POS tags of the Chinese corpus [Zhou, 1996] and (2) the 1428 semantic clusters of the Chinese corpus, which were taken from "同义词词林" (TongYiCi CiLing), a widely used Chinese thesaurus [Mei, 1983]. As shown in Table 1, the perplexity was drastically decreased by increasing the number of word clusters. The best results on both Chinese and Japanese corpora are still the word-based trigram values. It turns out that human defined clusters work much worse than automatically derived clusters with similar numbers of word clusters. The results are consistent with those of Ney *et al*. [1994], who observed that for small amounts of training data (100,000 words), hand clustering outperformed automatic clustering, but that for larger amounts (1.1 million words), automatic

clustering was better.

Notice that although the perplexity of the hand clustering model is much higher than the perplexity of the automatic clustering model, this does not mean that human defined clusters are unreasonable or worse than automatically derived clusters. The two cluster sets were generated by different criteria and motivations. Hand clustering is usually based on semantic/syntactic similarity, while automatic clustering uses the perplexity measurement directly. Therefore, the former is more widely used for knowledge systems, such as spoken language understanding, while the latter is good for statistical systems, such as speech recognition. As shown in table 4, although most of the automatically derived clusters look reasonable, there are also clusters which are difficult to interpret from a linguistic point of view.

Table 1. *Test set perplexities with cluster-based trigram models.*

Number of clusters	Chinese	Japanese
2^5	644.31	346.05
2^6	542.13	268.92
2^7	464.76	223.70
2^8	405.92	194.26
2^9	358.57	172.63
2^10	322.13	155.39
28 (POS clusters)	1038.56	------
1428 (semantic clusters)	676.87	------
Word trigram	242.74	106.33

5.2.2 Using combined models

In the second series of experiments, we used the combined models of Equation (9), where the cluster-based trigram model is linearly interpolated with the word-based trigram model. The interpolation constant λ is optimized on heldout data. The results are shown in Table 2. We still used word-based trigram models as baseline systems. It turns out that combined models consistently outperform baseline models. Unlike the case shown in the Table 1, the perplexity is decreased slowly at first by increasing the number of word clusters. We thus have an optimum at about 2^9 clusters for both the Chinese and the Japanese corpus. Beyond these numbers, the perplexity increases slightly again. Depending on the corpus, we have different

levels of perplexity reduction: about 3% for the Chinese corpus (at 2^9 clusters), and more than 10% for the Japanese corpus (at 2^9 clusters).

Table 2. *Test set perplexities with combined trigram models.*

Number of clusters	Chinese	Japanese
2^5	236.01	100.01
2^6	235.02	98.21
2^7	234.21	96.68
2^8	233.53	95.73
2^9	233.42	95.41
2^10	234.11	95.66
2^11	234.81	96.72
2^12	235.53	97.60
2^13	236.58	99.58
Word trigram	242.74	106.33

5.2.3 Using higher-order *n*-gram models

While trigram approximation has been proven, in practice, to be reasonable, there is disagreement about whether longer contexts can be helpful. This has led to research on using *n*-gram models in which *n*>3, called higher-order *n*-grams. Most of the previous experiments with higher-order *n*-grams showed little improvement because of the data sparseness problem. For example, [Goodman, 2001] showed that even using a very large corpus for *n*-gram model training (e.g. 280 million words), very small improvements occurred for *n*-gram models, where *n* is larger than 5. Clustering is an alternative way of dealing with the data sparseness problem besides smoothing. It was, thus, interesting to explore the effectiveness of cluster-based higher-order *n*-gram models.

We performed the third series of experiments on the relationship between cluster-based *n*-gram order and perplexity. We fixed the number of clusters at 2^8, and built a series of *n*-gram models, with *n* ranging from 2 to 20. The cluster-based higher-order *n*-gram models were then linearly interpolated with normal word-based trigram models. The perplexity results are shown in Table 3. We can see that although we used training corpora of medial size, improvement still occurred even for very high order *n*-gram models. After 10-gram models were used, depending on the corpus, we obtained approximately 10% perplexity reduction for

the Chinese corpus, and obtained more than 11% perplexity reduction on the Japanese corpus. It then turns out that clustering works significantly better with higher-order *n*-gram models than the traditional smoothing methods as described in [Chen and Goodman, 1999].

Table 3. *Test set perplexities with cluster-based higher-order* n-*gram models.*

Order of cluster-based *n*-gram model	Chinese	Japanese
2	238. 93	99. 00
3	233. 53	95. 73
4	230. 17	93. 74
5	226. 79	93. 62
6	224. 52	93. 91
7	223. 01	94. 19
8	221. 37	94. 33
9	220. 05	94. 47
10	219. 44	94. 47
12	219. 14	94. 53
20	219. 13	94. 53
word trigram	242. 74	106. 33

5.2.4 Analysis of words in clusters

We divided the 50,180-entry Chinese lexicon into 2^8 clusters by means of automatic clustering. The number of words in each cluster varied greatly from 0 to more than 2000. Table 4 gives 11 examples of word clusters. For each cluster from **A** to **C**, we randomly selected 10 two-character Chinese words, and removed those words that occurred less than 10 times in the training corpus. For each remaining cluster shown in table 4, we give the top 15 to 30 two-character Chinese words with the highest frequency (at lest 10 times) in the training corpus.

We can see that most of the words in each cluster belong to the same syntactic class, namely, verbs for cluster **A**, nouns for clusters **B** and **C**, etc. Furthermore, there are some semantic similarities between the words in a cluster. The majority of the words in cluster **A** are verbs expressing some kind of motion, some of the words in cluster **B** are titles, and some of the words in cluster **C** are games. There are also words which appear to be in the wrong

cluster: words like “earth” and “banquet” are not games, and words like “parsimony” and “mournful” are not verbs. Although most of the clusters look reasonable, there are also clusters that are difficult to interpret from a linguistic point of view. The other 8 clusters, which contain only high frequency words, look quite reasonable. It turns out that, given a sufficient large training corpus, the degree to which the clusters capture both syntactic and semantic aspects of Chinese is quite impressive although they were constructed from nothing more than counts of bigrams.

Table 4. *Most frequent words of some sample clusters from the Chinese corpus.*

Cluster	Words
A	走(walk), 飞跑(run), 奔腾(rush), 高攀(climb up), 推翻(overset), 跳动(jump), 流淌(flow), 吝惜(parsimony), 凄然(mournful), …
B	老师(teacher), 先生(sir), 小姐(miss), 同志(comrade), 父亲(father), 母亲(mother), 讨伐(crusade against), 发誓(promise), …
C	篮球(basketball), 棒球(baseball), 乒乓球(ping-pang), 铅球(shot), 地球(earth), 宴会(banquet), …
D	进行(conduct), 建立(build), 提出(bring forth), 实现(accomplish), 取得(gain), 提供(provide), 出现(advent), 得到(annex), 形成(form), 发生(occur), 发挥(develop), 产生(accrue), 完成(complete), 获得(get), 发表(publish), 创造(create), 召开(convene), 出席(attend), 所有(all), …
E	继续(keep on), 再次(once more), 重新(over again), 坚决(determined), 第一次(the first time), 多次(several times), 经常(often), 纷纷(one after another), 突然(suddenly), 立即(at once), 刚刚(a moment ago), 逐渐(gradually), 尽快(as soon as possible), 主动(active), 从中(from it), 亲自(personally), 彻底(thoroughly), 提前(advanced), 反复(again and again), 马上(immediately), …
F	汽车(automobile), 石油(petroleum), 建筑(architecture), 制造(manufacture), 加工(process), 食品(food), 化学(chemistry), 化工(chemical engineering), 机械(mechanics), 本地(native), 广告(advertisement), 航空(aerial), 制作(manufacture), 航天(spaceflight), 示范(demonstration), 电力(power), 服装(garment), 纺织(spinning), 钢铁(steel and iron), 走私(smuggle), …

G	重要(important), 主要(dominant), 群众(mass), 一定(certain), 基本(elementary), 重大(fatal), 实际(practical), 一切(the whole), 高度(high), 人类(mankind), 一般(general), 具体(concrete), 根本(basic), 自然(natural), 核心(kernel), 特殊(special), 自身(oneself), 客观(objective), 各自(respective), 唯一(unique), 最好(best), 自我(self), 周围(surrounding), 军人(soldier), 绝对(absolute), 历史性(historic), 彼此(one another), 最低(lowest), …
H	广州(Guangzhou), 贫困(poor), 深圳(Shenzhen), 天津(Tientsin), 纽约(New York), 南京(Nanking), 厦门(Xiamen), 重庆(Chongqing), 巴黎(Paris), 东北(northeast), 西安(Xi'an), 福州(Fuzhou), 长江(Yangtze River), 华盛顿(Washington), 东京(Tokyo), 成都(Chengdu), 大连(Dalian), 珠海(Zhuhai), 武汉(Wuhan), 沿海(coastal), 西南(southwest), 南方(south), 黄河(Yellow River), 整顿(put to order), 山区(mountain region), …
I	所谓(so-called), 称为(call), 誉为(call), 可谓(call), 叫做(call), 称之为(call), 致函(address a letter), 评为(call), 人称(namely), 发给(hand out), 称作(call), 素有(have), 号称(reputed), 鼓吹(promote by publicity), 当作(regard as), …
J	过去(past), 以后(after), 之后(later on), 当时(at that time), 一天(one day), 后来(later on), 如今(now), 为此(for the purpose), 另外(for the purpose), 当年(that year), 晚上(night), 不久(soon), 面前(in front), 之前(before), 身上(body), 这时(this time), 拒绝(reject), 中间(intermediate), 随后(later on), 那天(that day), …
K	积极(positively), 不断(constantly), 充分(adequately), 认真(seriously), 广泛(widely), 深入(deeply), 正确(correctly), 有效(availably), 真正(really), 逐步(stepwise), 健康(healthily), 明显(obviously), 迅速(promptly), 严格(sternly), 明确(explicitly), 顺利(smoothly), 普遍(generally), 热烈(warmly), 热情(passionately), 合理(reasonably), 及时(timely), 切实(practically), 更好(get better), 有力(strongly), 大大(greatly), 显著(significantly), 自觉(voluntarily), 相应(correspondingly), …

5.3 Clustering for language model compression

As shown in the last subsection, Equation (9) leads to better results (lower perplexities) than a simple trigram model does. But at the same time, the combined model is larger, since it includes both a cluster-based trigram model and a normal trigram model. In this subsection, we will explain how we took memory constraints into consideration, and concentrated our experiments on using clustering for language model compression. We performed experiments on the three basic cluster-based trigram models described in Section 3, and we found that our novel clustering techniques could be combined with language model pruning methods to

produce much smaller models at a given level of perplexity than could be produced using pruning methods without clustering.

Since we are seeking the correct balance between memory storage and perplexity, all experimental results below are presented in the form of size/perplexity curves. The size was measured as the total number of parameters of the language model: one parameter for each bigram and trigram one parameter for each normalization parameter α that was needed, and one parameter for each unigram. In the pruning experiments, bigrams and trigrams were both pruned, unigrams never were. This resulted in the smallest possible number of parameters being equal to the vocabulary size, e.g. 50,187 unigrams for Chinese models, and 180,187 unigrams for Japanese models.

In our experiments described below, we used Stolcke's [1998] pruning method to produce a series of language models of different sizes. This method is an entropy-based cutoff method, and can be considered an extension of the work of Seymore and Rosenfeld [1996] and of Kneser [1996]. The basic idea is to remove as many "useless" probabilities as possible, and at the same time to keep the perplexity increase as small as possible. This is achieved by examining the weighted relative entropy or Kullback-Leibler distance between each probability $P(w|h)$ and its value from the backoff distribution, $\overline{P}(w|\overline{h})$:

$$D(P(w|h)\,||\,\overline{P}(w|\overline{h})) = P(w|h) * \log\frac{P(w|h)}{\overline{P}(w|\overline{h})}, \tag{18}$$

where $\overline{h}$ is the reduced history. When the Kullback-Leibler distance is small, the backoff probability is a good approximation, and the probability $P(w|h)$ does not carry much additional information and can be deleted. The Kullback-Leibler distance was calculated for each *n*-gram entry, and we removed entries and reassigned the deleted probability mass to backoff mass for any *n*-gram entry whose deletions increased the Kullback-Leibler distance by less than a specified threshold value. Compared to the traditional count cutoff methods, Stolcke pruning performed a little better [Goodman and Gao, 2000]. More importantly, the Stolcke method could prune a model to a specific size, simply by finding the threshold level that resulted in a model of that size. For all the models, we used a smoothing method called *modified absolute discounting* for backoff. We give more details about Stolcke pruning and modified absolute discounting in Appendix A.

We then performed a number of experiments to compare our different models. In these experiments, the baseline system was the word-based trigram model.

5.3.1 Predictive clustering

We first used predictive clustering of Equation (10). The results are shown in Figures 1 and 2. It turns out that we got the best result at about 2^6 clusters for both the Chinese and Japanese corpora. Depending on the corpus, compared to the baseline systems, at the same size, we got

a maximum 6.6% perplexity reduction for the Chinese corpus, and a maximum 5.1% perplexity reduction for the Japanese corpus; at the same perplexity, we got a maximum 54% size reduction for the Chinese corpus, and a maximum 57% size reduction for the Japanese corpus. We notice that for these two corpora, although we got the best result at 2^6 clusters for both of them, the results at other numbers of clusters (e.g. 2^4, 2^7) were very different. For the Chinese corpus, all the predictive clustering models performed about the same. For the Japanese corpus, models at larger numbers of clusters performed much better than models at small numbers of clusters (e.g. 2^4). In general, with our clustering, when there was only a small amount of training data, the clusters became less useful. Perhaps this was because there was a more serious data sparseness problem for the Chinese corpus, and many parameters were out of training, thus larger clusters did not bring benefits. As for the Japanese corpus, the data sparseness problem was much less serious, so a large number of clusters led to significant perplexity reduction.

We also tried to set different pruning threshold values for the two components of the predictive clustering models. We could not obtain any improvement. Therefore, in what follows, we will always assume that we used the same pruning threshold value for both components in the predictive clustering and combined clustering models.

5.3.2 Conditional clustering

We used the conditional clustering of Equation (11). As shown in Figures 3 and 4, the results for the two languages are qualitatively very similar. The performance was consistently improved by increasing the number of clusters. But no conditional clustering model was superior to the baseline model. This is not surprising because the conditional clustering model always discards information for predicting words, and even with smoothing, it does not bring any additional benefits.

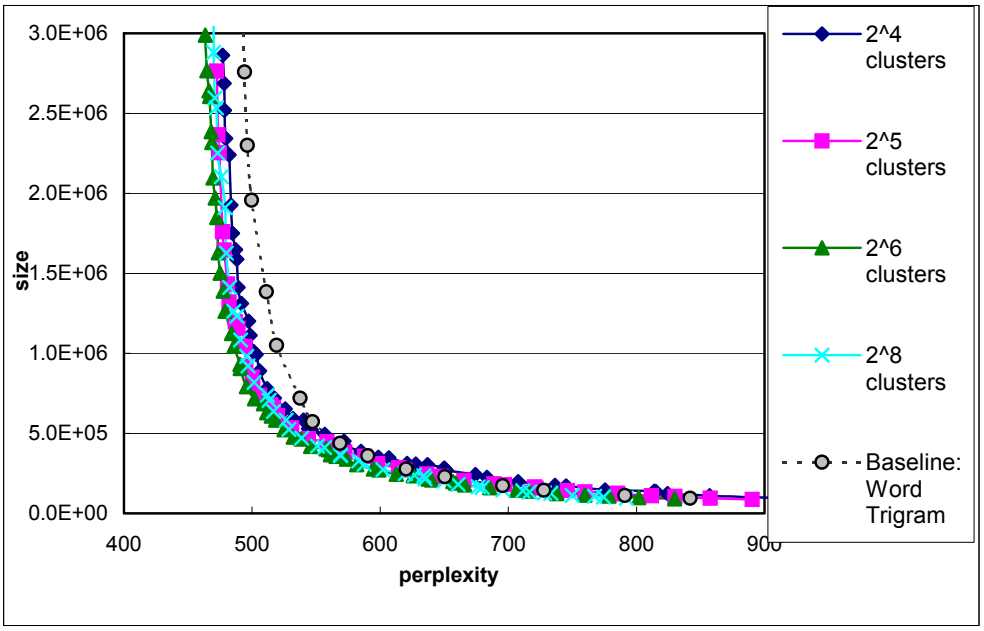


Figure 1 *Comparison of predictive models applied with different numbers of clusters to the Chinese corpus.*

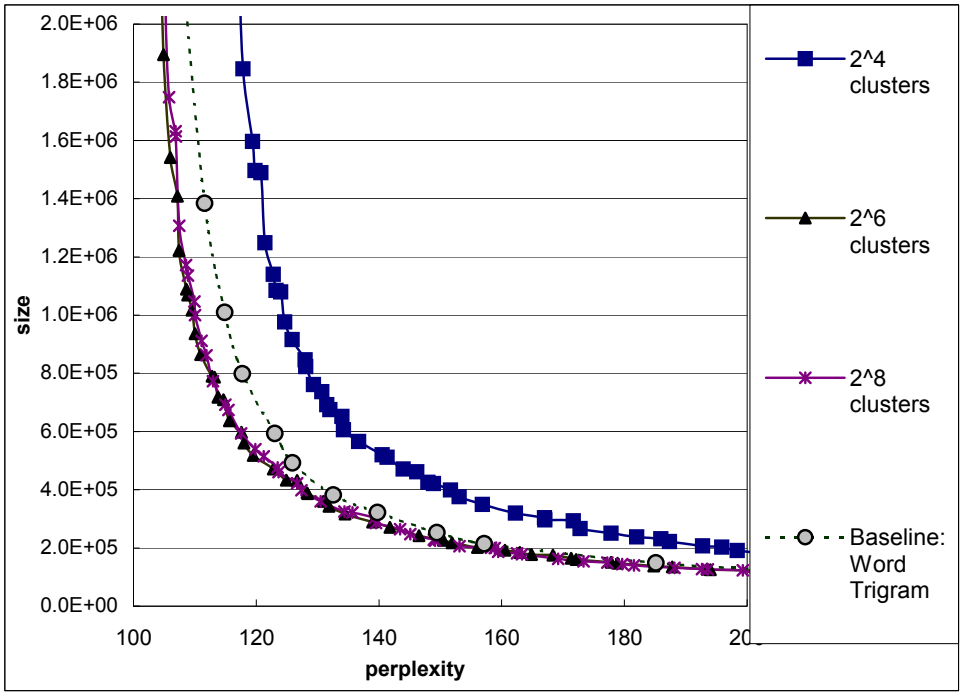


Figure 2. *Comparison of predictive models applied with different numbers of clusters to the Japanese corpus.*

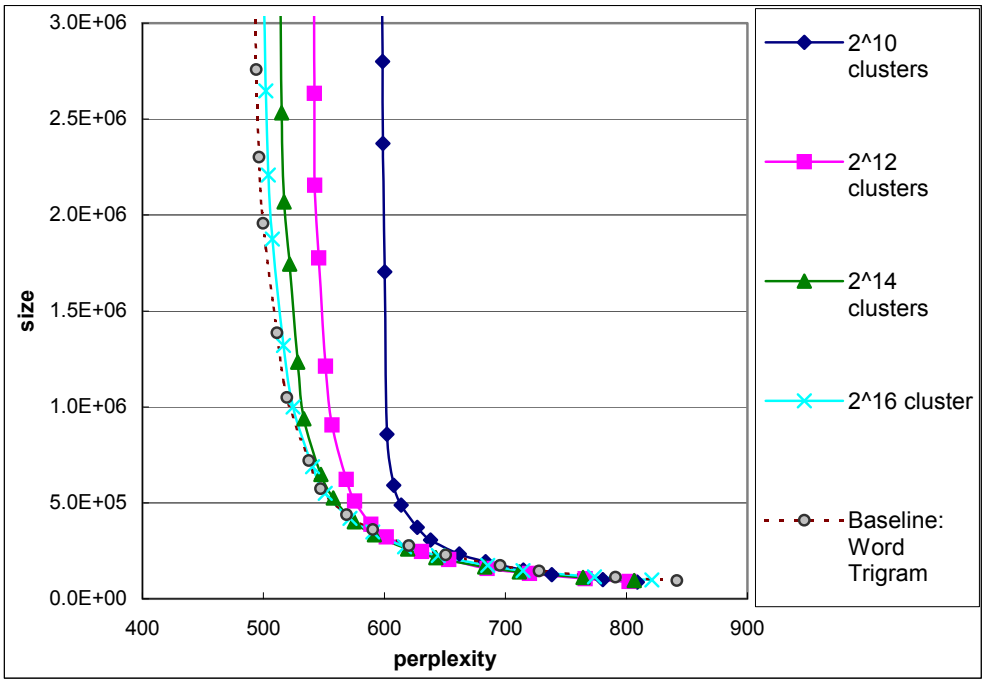


Figure 3 *Conditional models applied with different numbers of clusters to the Chinese corpus.*

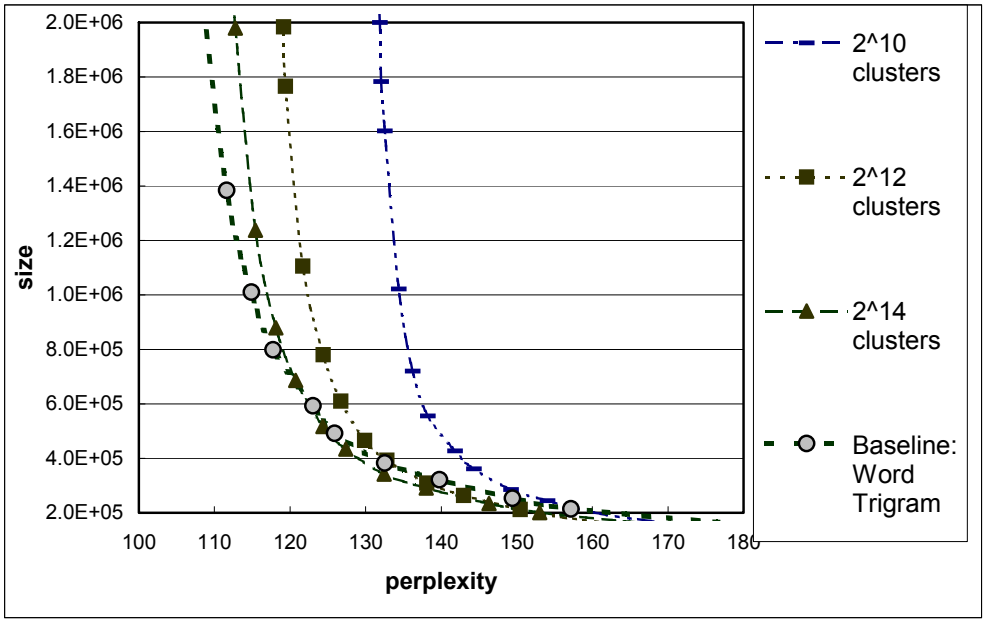


Figure 4 *Comparison of conditional models applied with different numbers of clusters to the Japanese corpus.*

5.3.3 Combined clustering

We also used the combined clustering of Equation (12). As mentioned earlier, we can use different numbers of cluster for predictive clusters and conditional clusters. This leads to a very large number of possible parameter settings. We presented detailed analysis of the parameter settings of the combined clustering model in [Goodman and Gao, 2000]. In this paper, we will only report the results of some sample parameter settings.

For the Chinese corpus, as shown in Figure 5, we set the number of predictive clusters to 2^4, 2^6, and 2^8, and set the number of conditional clusters to 2^12, 2^14, and 2^16. We then built a large number of models. Rather than graph all the points of all the models, we show only the outer envelope of the points for each number of predictive clusters in Figure 5. That is, if for a model with a given number of predictive clusters, there was some other point with the same number of predictive clusters (and perhaps a different number of conditional clusters) with both lower perplexity and smaller size than the first model, then we did not graph the first, worse point. We show the outer envelope of the size/perplexity curves of 2^4, 2^6, and 2^8 predictive clusters.

For the Japanese corpus, as shown in Figure 6, we do not show the outer envelopes as in Figure 5. Instead, we show results of the top three best parameter settings we obtained; for instance, (2^4, 2^12) represents the combined cluster-based trigram model with 2^4 predictive clusters and 2^12 conditional clusters.

It turns out that, for the Japanese corpus, the best combined clustering models outperformed the baseline model. At small model sizes, we got the best result at 2^14 conditional clusters and 2^6 predictive clusters. At large model sizes, we got the best result at 2^12 conditional clusters and 2^4 predictive clusters. We achieved the maximum 6.5% perplexity reduction at the same size, and the maximum 40% size reduction at the same perplexity. But for the Chinese corpus, no improvement over the baseline model was achieved until we used models with very large numbers of conditional clusters. This is not difficult to explain. Recall that predictive clustering is a special case of combined clustering. Actually, in most combined clustering models for Chinese, it turns out to be no less optimal to use conditional clusters than the vocabulary size, i.e., no conditional clustering.

Now, consider the IBM clustering of Equation (6), which is a special case of the combined clustering model. As shown in Figure 6, the performance is by far the worst, roughly an order of magnitude worse than the performance of the other approaches. We hypothesized that this was because the IBM model throws away too much useful information. We thus tried a variation on the IBM model:

$$\lambda P(w_i \mid w_{i-2}w_{i-1}) + (1-\lambda)P(w_i \mid c_{i-2}c_{i-1}c_i) \times P(c_i \mid c_{i-2}c_{i-1}) . \quad (20)$$

This model is just like the standard IBM model, but it also conditions the probability of the word on the previous clusters. We compared this model with a standard IBM model. The results are shown in Tables 5 and 6. It turns out that, for the Chinese corpus, models in the form of Equation (20) consistently outperformed the standard IBM models (e.g. we achieved 4% perplexity reduction at 2^9 clusters), while for the Japanese corpus, they worked about the same. Notice that in these experiments, no pruning was done.

We summarize the results of all the experiments described in this subsection in Figures 7 and 8. It is clearly seen that our novel clustering techniques produce much smaller models than do previous methods (i.e. baseline systems) at the same perplexity level. In addition, several more conclusions can be drawn:

1. Conditional clustering did not help for either the Chinese or the Japanese corpus since it always discards information.
2. For closed tests on homogeneous text corpora (e.g. the Japanese corpus), both combined clustering and predictive clustering outperformed the baseline system consistently. Combined clustering is better at small model sizes, while predictive clustering is better at larger sizes.
3. For open tests on heterogeneous text corpora (e.g. the Chinese corpus), predictive clustering outperformed the baseline system consistently. Although the results presented in this paper show that combined clustering achieved no improvement, in [Goodman and Gao, 2000], we showed that with more sophisticated techniques, it appears possible to make combined clustering better than predictive clustering.

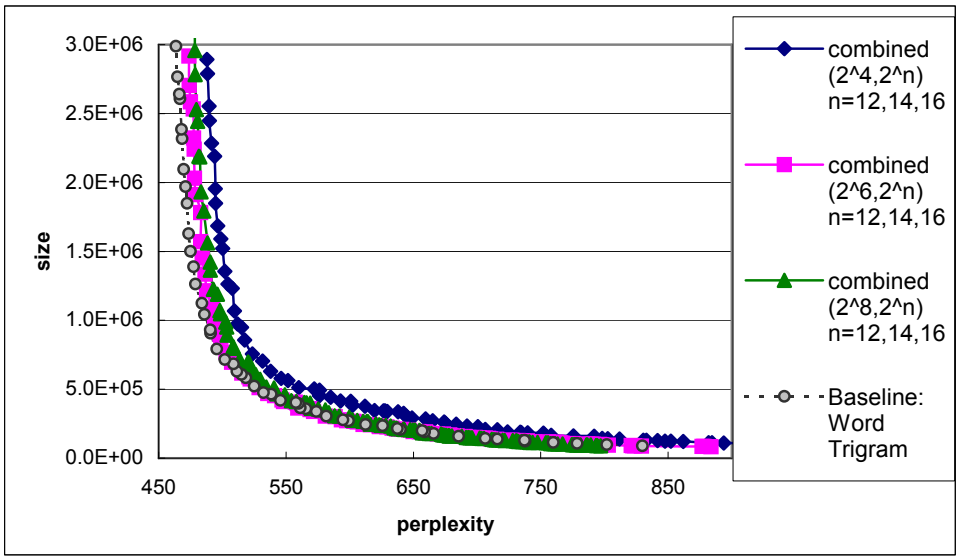


Figure 5 *Comparison of combined clustering models applied with different numbers of clusters to the Chinese corpus.*

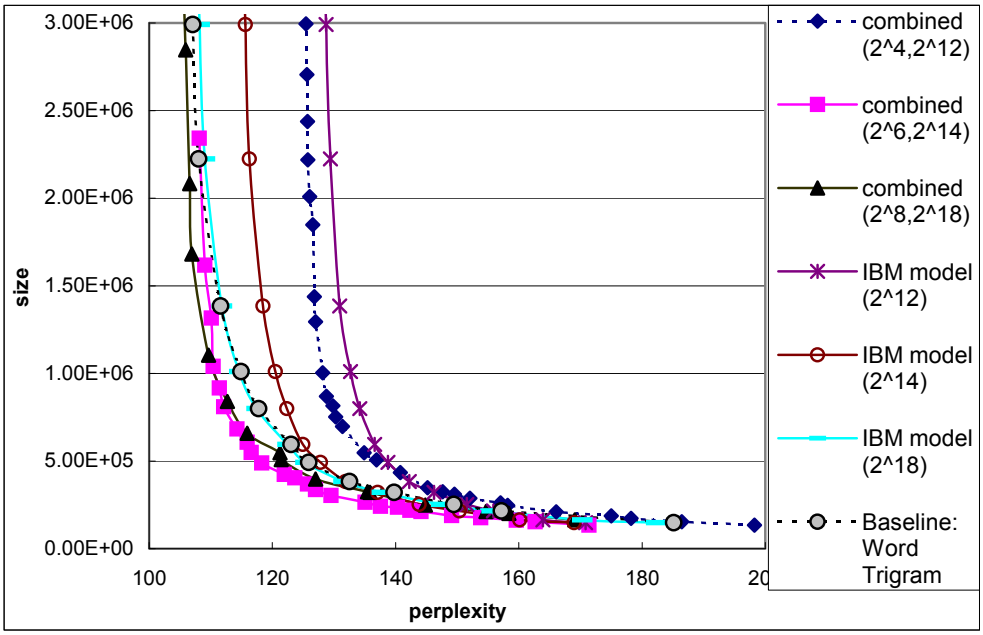


Figure 6 *Comparison of combined clustering models applied with different numbers of clusters to the Japanese corpus and the IBM model.*

Table 5. *Comparison of different combined trigram models applied to the Chinese corpus.*

Number of clusters	Equation (9)	Equation (20)
2^6	235. 02	226.65
2^7	234. 21	224.65
2^8	233. 53	224.29
2^9	233. 42	224.99
2^10	234. 11	226.53
2^11	234. 81	228.26
2^12	235. 53	230.95
2^13	236. 58	234.78
word trigram	242. 74	242.74

Table 6. *Comparison of different combined trigram models applied to the Japanese corpus.*

Number of clusters	Equation (9)	Equation (20)
2^6	98. 21	97.06
2^7	96. 68	96.42
2^8	95. 73	96.33
2^9	95. 41	96.41
2^10	95. 66	96.82
2^11	96. 72	97. 57
2^12	97. 60	98. 50
2^13	99. 58	100. 52
word trigram	106. 33	106. 33

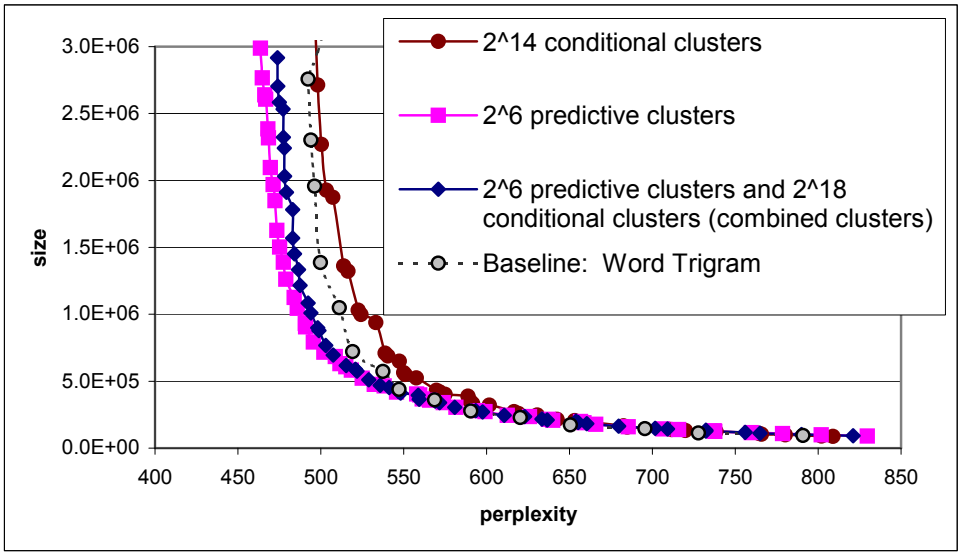


Figure 7 *Summary of the results obtained by applying clustering models to the Chinese corpus.*

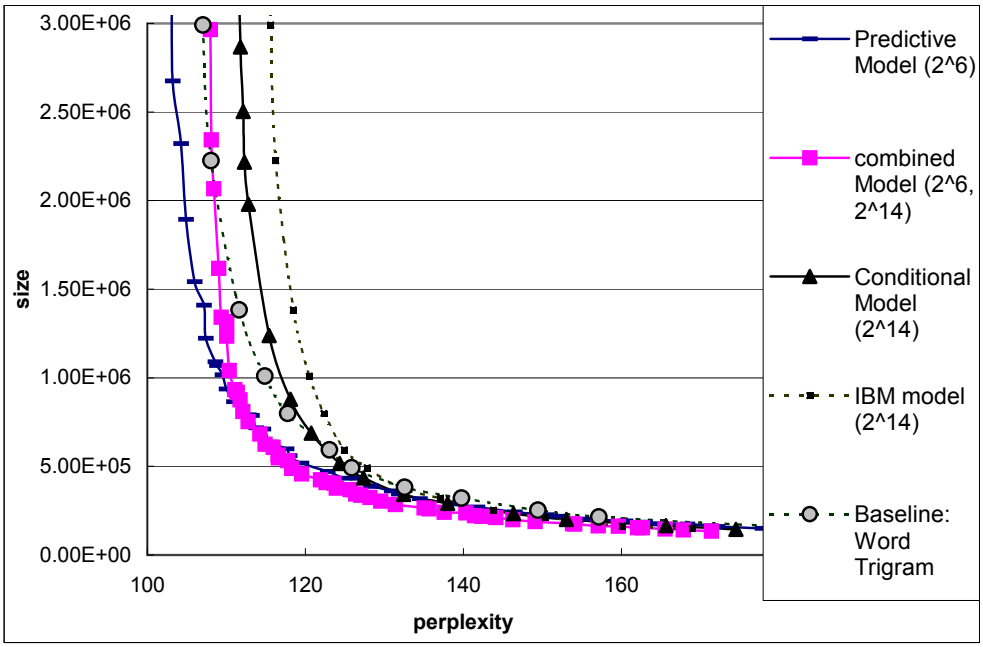


Figure 8 *Summary of the results obtained by applying clustering models to the Japanese corpus.*

6. Conclusion

Cluster-based *n*-gram models are variations on traditional word-based *n*-gram models. They attempt to make use of the similarities between words. In this paper, we have presented an empirical study on clustering techniques for Asian language modeling. While the majority of the previous research on word clustering has focused on how to get the best clusters, we have concentrated our research on the best way to use the clusters. We have studied in detail three cluster-based *n*-gram models, namely, *predictive clustering*, *conditional clustering*, and *combined clustering*. In our experiments, clustering was used to improve the performance (i.e. perplexity) of language models as well as to compress language models. We performed experimental tests on a Japanese newspaper corpus of more than 10 million words, and on a Chinese mixed-domain corpus of more than 7 million words. Results show that our novel techniques worked much better than previous methods. They not only showed better performance when interpolated with normal *n*-gram models, but could also be combined with Stolcke pruning to produce models much smaller than unclustered models with the same perplexity level.

Most language modeling improvements reported previously required significantly more space than the normal trigram baseline model, or had higher perplexity. Their practical value

is questionable. In this paper, we have proposed a technique that results in lower perplexity than the traditional trigram models do at every memory size. In other research [Gao *et al*., 2001], we have shown that cluster-based models of this form can be applied effectively to pinyin to Chinese character conversion. One area we consider promising for future research is the combination of human defined and automatically derived clustering. While human defined clusters alone generally work worse than automatically derived clusters, there has been little research on their combination. It is an open question whether such a combination can lead to further improvements.

Acknowledgements

We would like to thank Prof. Changning Huang, Dr. Ming Zhou, and other colleagues at Microsoft Research, Yoshiharu Sato, and Hiroaki Kanokogi at the Microsoft (Japan) IME group, for their help in developing the ideas and implementation presented in this paper. We would also like to thank Jiang Zhu and Miyuki Seki for their help in our experiments and for providing Chinese and Japanese text corpora.

References

Bai, S., Li, H., Lin, Z., and Yuan, B. "Building class-based language models with contextual statistics." In *ICASSP-98*, 1998. pp. 173-176.

Bellegarda, J. R., Butzberger, J. W., Chow, Y. L., Coccaro, N. B., and Naik, D. "A novel word clustering algorithm based on latent semantic analysis." In *ICASSP-96*, 1996. pp. I172-I175.

Brown, P. F., Cocke, J., DellaPietra, S. A., DellaPietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. "A statistical approach to machine translation." *Computational Linguistics,* 16(2), 1990. pp. 79-85.

Brown, P. F., DellaPietra V. J., deSouza, P. V., Lai, J. C., and Mercer, R. L. "Class-based n-gram models of natural language." *Computational Linguistics,* 18(4), 1992. pp. 467-479.

Chen, S. F., and Goodman, J. "An empirical study of smoothing techniques for language modeling." *Computer Speech and Language,* 13:359-394, October. 1999.

Church, K. "A stochastic parts program and noun phrase parser for unrestricted text." In *Proceedings of the Second Conference on Applied Natural Language Processing*, 1988. pp. 136-143.

Cutting, D. R., Karger, D. R., Pedersen, J. R., and Tukey, J. W. "Scatter/gather: A cluster-based approach to browsing large document collections." In *SIGIR 92*. 1992.

Gao, J., Goodman, J., Li, M., and Lee, K. F. "Toward a unified approach to statistical language modeling for Chinese." To appear in *ACM Transactions on Asian Language*

Information Processing. 2001. Draft available from http://www.research.microsoft.com/~jfgao

Goodman, J. "A bit of progress in language modeling." Submitted to *Computer Speech and Language*. 2001. Draft available from http://www.research.microsoft.com/~joshuago

Goodman, J., and Gao, J. "Language model compression by predictive clustering." *ICSLP-2000*, Beijing, October. 2000.

Heeman, P. "POS tags and decision trees for language modeling." In *ACL-99*, 1999. pp. 129-137.

Heeman, P., and Allen, J. "Incorporating POS tagging into language modeling." In *Eurospeech-97*, Ghodes, Greece, 1997. pp. 2767-2770.

Huang, X. D., Acero, A., and Hon, H. Spoken language processing. Prentice Hall PTR. 2001.

Issar, S., and Ward, W. "Flexible parsing: CMU's approach to spoken language understanding." In *Proceedings of the ARPA Spoken Language Technology Workshop*, pp. 53-58. 1994.

Jelinek, F. Self-organized language modeling for speech recognition. In *Readings in Speech Recognition*, A. Waibel and K. F. Lee, eds., Morgan-Kaufmann, San Mateo, CA, 1990, pp. 450-506.

Jurafsky, D., and Martin, J. H. Speech and language processing. Prentice Hall.

Katz, S. M. 1987. "Estimation of probabilities from sparse data for the language model component of a speech recognizer." *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35(3):400-401, March. 2000.

Kernighan, M. D., Church, K. W., and Gale, W. A. "A spelling correction program based on a noisy channel model." In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, 1990. pp. 205-210.

Kneser, R. and Ney, H. "Improved clustering techniques for class-based statistical language modeling." In *Eurospeech,* Vol. 2*,* 1993. pp. 973-976, Berlin, Germany.

Kneser, R. "Statistical language modeling using a variable context length." Proc. ICSLP, volume 1, pages 494-497, Oct. 1996.

Maltese, B., and Mancini, F. "An automatic technique to include grammatical and morphological information in a trigram-based statistical language model." In *ICASSP-92*, 1992. pp. I157-I160.

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

Mei, J. Z. Tongyici Cilin. Shanghai Cishu Publishing House, 1983. Shanghai.

Miller, D., Leek, T., and Schwartz, R. M. "A hidden Markov model information retrieval system." In *Proc. 22nd International Conference on Research and Development in Information Retrieval*, Berkeley, CA, 1999, pp. 214-221.

Miller, J. W., and Alleva, F. "Evaluation of a language model using a clustered model back off." In *ICASSP-97*, 1997. pp. 390-393.

Ney, H., Essen, U., and Kneser, R. "On structuring probabilistic dependences in stochastic language modeling." *Computer Speech and Language*, 8:1-38. 1994.

Niesler, T. R., Whittaker, E. W. D., and Woodland, P. C. "Comparison of part-of-speech and automatically derived category-based language models for speech recognition." In *ICASSP-98*, 1998. pp. I177-I180.

Pereira, F., Tishby, N., and Lee L. "Distributional clustering of English words." In *Proceedings of the 31st Annual Meeting of the ACL.* 1993.

Placeway, P., Schwartz, R., Fung, P., and Nguyen, L. "The estimation of powerful language models from small and large corpora." In *ICASSP-93*, 1993. II33-36.

Seymore, K. and Rosenfeld, R. "Scalable backoff language models", *Proc. ICSLP*, Vol. 1., 1996. pp.232-235, Philadelphia,

Srinivas, B. "Almost parsing techniques for language modeling." In *ICSLP-96*, 1996. pp. 1169-1172.

Stolcke, A. "Entropy-based Pruning of Backoff Language Models." In *Proc. DARPA News Transcription and Understanding Workshop*, Lansdowne, VA. 1998. pp. 270-274. See corrections at http://www.speech.sri.com/people/stolcke

Ueberla, J. P. "An extended clustering algorithm for statistical language models." *IEEE Transactions on Speech and Audio Processing*, 4(4): 313-316. 1996.

Ward, W., and Young, S. "Flexible use of semantic constraints in speech recognition." In ICASSP-93, 1993. pp. II49-50.

Yamamoto, H., and Sagisaka, Y. "Multi-class Composite N-gram based on Connection Direction." In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, May, 1999. Phoenix, Arizona.

Yang, Y. J., *et al*., "An intelligent and efficient word-class-based Chinese language model for Mandarin speech recognition with very large vocabulary." In *ICSLP-94*, Yokohama, Japan, 1994. pp. 1371-1374.

Zhou, Q. "Phrase bracketing and annotating on Chinese language corpus." Ph.D. dissertation. Beijing University. 1996.

Zue, V. W. Navigating the information superhighway using spoken language interfaces. *IEEE Expert*, vol. 10, no. 5, pp. 39-43, October, 1995

A. Methods of Trigram Training

We will describe methods for language model training below. These include the modified absolute discounting smoothing method and Stolcke's entropy-based pruning method.

Absolute Discounting

Trigram Language models make the assumption that the probability of a word depends only on the identity of the immediately two preceding words, say $P(w_i|w_1\ w_2 ...\ w_{i-1}) \approx P(w_i|w_{i-2}\ w_{i-1})$.

Smoothing is used to address the problem of data sparseness. Experimental results show that a novel variation of absolute discounting, Kneser-Ney smoothing, consistently outperforms all others [Chen and Goodman, 1999]. However, because Kneser-Ney smoothing is less commonly used, slightly more difficult to implement, and, we suspect, may not work as well when pruning is done, we used a slightly different technique in this research, modified absolute discounting. First, we will describe basic absolute discounting. Letting D represent a discount, we set the probability as follows:

$$P_{absolute}(w_i \mid w_{i-2}w_{i-1}) = \begin{cases} \dfrac{\max(C(w_{i-2}w_{i-1}w_i) - D, 0)}{C(w_{i-2}w_{i-1})} & \text{if } C(w_{i-2}w_{i-1}w_i) > 0 \\ \alpha(w_{i-2}w_{i-1}) P_{absolute}(w_i \mid w_{i-1}) & \text{otherwise} \end{cases} . \quad (21)$$

$\alpha(w_{i-2}w_{i-1})$ is defined in such a way that the probabilities sum to 1:

$$\alpha(w_{i-2}w_{i-1}) = \frac{1 - \sum_{w_i : C(w_{i-2}w_{i-1}w_i) > 0} P_{absolute}(w_i \mid w_{i-2}w_{i-1})}{1 - \sum_{w_i : C(w_{i-2}w_{i-1}w_i) > 0} P_{absolute}(w_i \mid w_{i-1})} . \quad (22)$$

The trigram backs off to the bigram, and the bigram backs off to the unigram. The unigram does not need to be smoothed although it can be smoothed with the uniform distribution. In practice, different D's are used for the bigram and trigram.

A further improvement is to use multiple discount D. Taking the trigram as an example, D_1 stands for counts $C(w_{i-2}w_{i-1}w_i) = 1$, D_2 for $C(w_{i-2}w_{i-1}w_i) = 2$, and a final one, D_3 for $C(w_{i-2}w_{i-1}w_i) \geq 3$. Chen and Goodman [1999] introduced an estimate for the optimal D for absolute discounting smoothing as a function of training data counts[1]. In practice, we can use Equation (23) to Equation (26) to approximately estimate the optimal values for D_1, D_2, and

[1] Thanks to Ries, K.

D_3:

$$Y = \frac{n_1}{n_1 + 2n_2}, \tag{23}$$

$$D_1 = 1 - 2Y\frac{n_2}{n_1}, \tag{24}$$

$$D_2 = 2 - 3Y\frac{n_3}{n_2}, \tag{25}$$

$$D_3 = 3 - 4Y\frac{n_4}{n_3}. \tag{26}$$

where n_1, n_2, n_3, and n_4 are total numbers of trigrams with exactly one, two, three, and four counts, respectively.

Notice that for the experiments conducted in this study, we did not use this approximation, but instead optimized the discounts on heldout data. This led to very limited improvement.

Entropy-based Pruning

Stolcke [1998] proposed a criterion for pruning *n*-gram language models based on the relative entropy between the original and the pruned model. The relative entropy measure can be expressed as a relative change in training data perplexity. All *n*-grams that change perplexity by less than a threshold are removed from the model.

Formally, let P denote the trigram probabilities assigned by the original model, say $P=P(w_i|w_{i-2}w_{i-1})$, and let $P'= P(w_i| w_{i-1})$ denote the probabilities in the pruned model, assuming that we have pruned the trigram probability. Then, the relative entropy between the two models is

$$\begin{aligned} D(P \| P') = -P(w_{i-2}w_{i-1})\{P(w_i \mid w_{i-2}w_{i-1})[\log P(w_i \mid w_{i-1}) + \log\alpha'(w_{i-2}w_{i-1}) - \log P(w_i \mid w_{i-2}w_{i-1})] \\ + [\log\alpha'(w_{i-2}w_{i-1}) - \log\alpha(w_{i-2}w_{i-1})] \sum_{w_i:C(w_i,w_{i-2}w_{i-1})=0} P(w_i \mid w_{i-2}w_{i-1})\} \end{aligned} \tag{27}$$

where $\alpha'(w_{i-2}w_{i-1})$ is the revised backoff weight after pruning. Recall that $\alpha(w_{i-2}w_{i-1})$ is estimated by Equation (22), and that $\alpha'(w_{i-2}w_{i-1})$ is obtained by dropping the term for the pruned trigram $(w_{i-2}w_{i-1}w_i)$ from the summation in both numerator and denominator.

B. Clustering Algorithm

There is no shortage of techniques for generating clusters, and there appears to be little

evidence that different techniques that optimize the same criterion result in significantly different quality of clusters. We note, however, that different algorithms may require significantly different amounts of run time. We used several techniques to speed up our clustering significantly.

The basic criterion we followed was to minimize entropy. In particular, we assumed that the model we were using was of the form $P(z|Y)$; we wanted to find the placement of words y into clusters Y that minimized the entropy of this model. This is typically done by swapping words between clusters whenever such a swap reduces the entropy.

The first important approach we took to speeding up clustering was a top-down approach. We note that agglomerative clustering algorithms – those which merge words bottom up – may require significantly more run time than top-down, splitting algorithms. Thus, our basic algorithm is top-down. However, in the end, we sometimes perform four iterations of swapping all words between all clusters. Notice that for the experiments reported in this paper, we used the basic top-down algorithm without swapping.

Another technique we used is Buckshot [Cutting *et al*., 1992]. The basic idea is that even with a small number of words, we are likely to have a good estimate of the parameters of a cluster. Therefore, we proceeded top down, splitting clusters. When we were ready to split a cluster, we randomly picked a few words, and put them into two random clusters, and then swapped them in such a way that entropy was decreased, until convergence occurred (no more decrease could be achieved). Then we added a few more words, typically $\sqrt{2}$ more, put each one into the best bucket, and then swapped again until convergence occurred. This was repeated until all words in the current cluster had been added and split. We haven't tested this particularly thoroughly, but our intuition is that it should lead to large speedups.

We used one more important technique that speeds up computations, adapted from an earlier work by [Brown *et al*., 1992]. We attempted to minimize the entropy of our clusters. Let v represent words in the vocabulary, and let W represent a potential cluster. We minimize

$$\sum_v C(Wv) \log P(v \mid W) .$$

The inner loop of this minimization considers adding (or removing) a word x to cluster W. What will the new entropy be? On its face, this would appear to require computation proportional to the vocabulary size to re-compute the sum. However, letting the new cluster $W + x$ be called X,

$$\sum_v C(Xv) \log P(v \mid X) = \sum_{v|C(xv)\neq 0} C(Xv) \log P(v \mid X) + \sum_{v|C(xv)=0} C(Xv) \log P(v \mid X) . \qquad (28)$$

The first summation in Equation 28 can be computed relatively efficiently, in an amount

of time proportional to the number of different words that follow x; it is the second summation that needs to be transformed:

$$\begin{aligned}\sum_{v|C(xv)=0} C(Xv)\log P(v\mid X) &= \sum_{v|C(xv)=0} C(Wv)\log\left(P(v\mid W)\frac{C(W)}{C(X)}\right)\\ &= \sum_{v|C(xv)=0} C(Wv)\log P(v\mid W)+\left(\log\frac{C(W)}{C(X)}\right)\sum_{v|C(xv)=0} C(Wv).\end{aligned} \tag{29}$$

Now, notice that

$$\sum_{v|C(xv)=0} C(Wv)\log P(v\mid W)=\sum_{v} C(Wv)\log P(v\mid W)-\sum_{v|C(xv)\neq 0} C(Wv)\log P(v\mid W), \tag{30}$$

and that

$$\sum_{v|C(xv)=0} C(Wv)=\left(C(W)-\sum_{v|C(xv)\neq 0} C(Wv)\right), \tag{31}$$

Substituting Equation 30 and 31 into Equation (29), we get

$$\begin{aligned}&\sum_{v|C(xv)=0} C(Xv)\log P(v\mid X)\\ &=\sum_{v} C(Wv)\log P(v\mid W)-\sum_{v|C(xv)\neq 0} C(Wv)\log P(v\mid W)+\left(\log\frac{C(W)}{C(X)}\right)\left(C(W)-\sum_{v|C(xv)\neq 0} C(Wv)\right).\end{aligned} \tag{32}$$

Now, notice that $\sum_{v} C(Wv)\log P(v\mid W)$ is just the old entropy, before adding x. Assuming that we have pre-computed/recorded this value, all the other summations only sum over words v for which $C(xv) > 0$, which, in many cases, is much smaller than the vocabulary size.

Many other clustering techniques [Brown *et al.*, 1992] attempt to maximize $\sum_{Y,Z} P(YZ)\log\frac{P(Y\mid Z)}{P(Z)}$, where the same clusters are used for both. The original speedup formula uses this version, and is much more difficult to minimize. Using different clusters for different positions not only leads to marginally lower entropy, but also leads to simpler clustering.

Computational Linguistics and Chinese Language Processing
Vol. 6, No. 1, February 2001, pp. 61-82

Locating Boundaries for Prosodic Constituents in Unrestricted Mandarin Texts

Min Chu*, Yao Qian+

Abstract

This paper proposes a three-tier prosodic hierarchy, including prosodic word, intermediate phrase and intonational phrase tiers, for Mandarin that emphasizes the use of the prosodic word instead of the lexical word as the basic prosodic unit. Both the surface difference and perceptual difference show that this is helpful for achieving high naturalness in text-to-speech conversion. Three approaches, the basic CART approach, the bottom-up hierarchical approach and the modified hierarchical approach, are presented for locating the boundaries of three prosodic constituents in unrestricted Mandarin texts. Two sets of features are used in the basic CART method: one contains syntactic phrasal information and the other does not. The one with syntactic phrasal information results in about a 1% increase in accuracy and an 11% decrease in error-cost. The performance of the modified hierarchical method produces the highest accuracy, 83%, and lowest error cost when no syntactic phrasal information is provided. It shows advantages in detecting the boundaries of intonational phrases at locations without breaking punctuation. 71.1% precision and 52.4% recall are achieved. Experiments on acceptability reveal that only 26% of the mis-assigned break indices are real infelicitous errors, and that the perceptual difference between the automatically assigned break indices and the manually annotated break indices are small.

1. Introduction

The state-of-the-art text-to-speech (TTS) systems are able to produce very natural synthesized

* Microsoft Research Asia, 5F, Beijing Sigma Center No. 49, Zhichun Road, Haidian District, Beijing 100080, P.R.C., E-mail: minchu@microsoft.com
+ Shanghai Normal University, 100 Guilin Road,Shanghai 200234, P.R.C., E-mail: yqian@shtu.edu.cn

speech if they are provided with a correct phonetic string and with prosodic features extracted from human pronunciation of the string [Chu and Lu, 1996; Dutoit *et al*., 1996]. Automatic prosody generators, however, cannot yet deliver high quality prosody. One of the main obstacles to automatic generation of prosody is the difficulty of identifying the hierarchical prosodic constituents from texts automatically. It has been proven through many experiments [Liberman and Prince, 1977; Gee and Grosjean, 1983; Selkirk, 1984; Ladd and Campbell, 1991] that prosody constituents are not always identical to those of the surface syntax. The relationship between prosody and syntax is not well understood. While, representing prosodic constituents by means of syntactic constituents directly cannot produce very natural prosody, the boundaries of prosodic constituents can be derived from syntactic information. Some early studies used rules to parse prosodic structures. Stochastic models have been used more frequently in recent studies. In some works [Wang and Hirschberg, 1991; Hirschberg and Prieto, 1996; Lee and Oh, 1999], break indices have been predicted using the automatic *classification and regression tree* (CART) from information such as four-word part-of-speech(POS) windows, pitch accent types, the sentence length, the distance from the beginning of the sentence and the end of the sentence, etc. A Markov model is used in works done by Veilleux *et al*. [Veilleux *et al*., 1990], which predicts the most likely sequence of break indices from the input POS sequence based on the assumption that the current index is only related to the previous index. Ostendorf and Veilleux [Ostendorf and Veilleux, 1994] proposed a hierarchical stochastic model for locating prosodic boundaries. Most of the publications on locating prosodic boundaries have focused on alphabet-based languages, such as English, which are very different from Mandarin in nature. Chou *et al*. [Chou *et al*., 1996; Chou *et al*., 1998] presented a top-down procedure for labeling break indices in Mandarin from both acoustic features, such as f_0, duration and energy, and features derived from text transcriptions. They reported that the acoustic features are helpful for predicting prosodic phrases. Since the prosodic boundary detecting approach presented in this paper is meant to be used in the Mandarin TTS system, where no acoustic features are available, only features that can be derived from text transcriptions will be used.

There are many reports specifying various hierarchical structures for prosodic constituents. The intonational phrase (INP) and the intermediate phrase (IMP) are the most commonly accepted levels in English. An English sentence consists of a sequence of INPs and each INP, in turn, is composed of a sequence of IMPs. INPs should have boundary tones at their ends, and IMPs are theoretically marked with phrase accents. Both types of phrases are cued by lengthening of the final syllables. With the above definition of prosodic hierarchy in English, studies have been done on predicting either one of the two prosodic phrases or both. The two prosodic constituents have been referred to as the major phrase and minor phrase in some papers. The word is used as the basic unit in all prosodic-phrase detecting algorithms in English.

Though Ostendorf and Veilleux [Ostendorf and Veilleux, 1994] mentioned the usefulness of considering the prosodic word (LW) rather than the lexical word (LW) as possible sites for break indices, they did not use them in their prosody model since it was difficult to define PWs relative to LWs. Most prosody related studies on Mandarin [Chou *et al*., 1996; Shen and Xu, 2000] have borrowed the two levels of prosodic phrases from English. In addition to the IMP and the INP, Chou *et al*. [Chou *et al*., 1998] defined a breath group boundary and a prosodic group boundary for short paragraphs. The two groups often contain more than one simple sentence. In this paper, only prosodic constituents smaller than sentences will be studied. Only the INP and the IMP are kept. However, our study shows that the PW word is a very important prosodic unit for Mandarin. The surface difference and perceptual difference between the PW and the LW will be introduced in Section 2. These differences show that using PWs instead of LWs as the basic unit of prosody will lead to improved naturalness of the synthesized speech. Thus, in our approach, a three-tier hierarchy is defined for prosody below the sentence level in Mandarin. The PW is the lowest constituent in the prosodic hierarchy. The middle tier is the IMP, which has a perceptive minor break at the end. The INP is the top tier with a major break at the end. The concepts of phrase accent and boundary tone in English are not easy to use in the definition of the IMP and the INP in Mandarin since Mandarin is a tonal language. The degree of break becomes the main cue for identifying them in real speech. The aim of this study was to locate the boundaries for the three-tier prosodic constituents automatically in unrestricted Chinese texts, using only information that can be derived from the texts.

The remainder of this paper is organized as follows. Section 2 discusses the surface difference and perceptive difference between the PW and the LW. Section 3 defines the three-tier prosodic constituents in Mandarin. Section 4 presents the three approaches to locating prosodic boundaries. Experiments and results are given in Section 5. Section 6 gives conclusions.

2. Prosodic Word vs. Lexicon Word

Since in many Asian languages, such as Chinese, Japanese or Korean, texts do not contain any visual cues for word boundaries, word segmentation becomes a basic requirement for almost all text analyses in these languages. Many studies had been done on word segmentation. Chinese has a very flexible list of words. The size of the lexicon used for word segmentation changes from 40,000 items to several hundreds of thousands of items. Most Chinese characters are words by themselves and also parts of longer words. The length of a word in characters ranges from 1 to 10 or more. However, in spoken Chinese, there exists a disyllabic rhythm. Succeeding mono-character words are often uttered as one disyllabic unit of rhythm, and long words are often uttered as several units. The unit of rhythm in Mandarin is referred as the prosodic word, which is defined as a group of syllables that should be uttered closely and continuously.

Although, in real speech, not all boundaries of PWs have breaks, it is tolerable if there is a break at the end of each PW. However, any inner PW break will make the speech unintelligible or unnatural. To distinguish then from PW, words listed in a lexicon used in word segmentation are referred to as lexical words. A PW may contain one or more LWs and it may also be only part of an LW. For example, in the Chinese sentence, "我买了一本好书 (I brought a good book)," each character itself is an LW. Yet, in natural speech, the sentence is grouped as "我\买了\一本\好书." There are four PWs. Since a PW is formed dynamically according to the context, many possible combinations of characters exist in real texts. It is impossible to list all the PWs in a lexicon as has been done for LWs. However, PW strings can be predicted from LW strings [Qian *et al*., 2001].

In an exploratory experiment, three annotators were asked to label the PW boundaries in 1348 utterances, with text transcriptions provided for these utterances. Table 1 lists the main guidelines for labeling PWs in speech. PW boundaries were labeled by both listening to the utterances and reading the text transcriptions. A 96.9% agreement ratio was achieved across three of them. The agreement ratio among at least two of them reached 99.9%. The high agreement ratio shows consistency in PW labeling across different people.

Table 1. *The main guidelines for labeling PWs by listening to the utterances and reading the text transcriptions.*

1.	A disyllabic or tri-syllabic LW is a PW if it has no proclitic or enclitic. Otherwise, it forms a PW with its clitic. Examples of enclitics are "的、了、着、（楼）上、（地）下、（物理）学、（革命）性"; examples of proclitics are "副(所长)、半(正式)."
2.	A mono-syllabic LW often forms a PW with the LW coming before or after it. Only when a mono-syllabic LW is lengthened enough to balance the disyllabic rhythm does it become a mono-syllabic PW.
3.	All LWs containing more than 3 syllables should be segmented into several disyllabic or tri-syllabic PWs according to their structures. When there are proclitics or enclitics, the clitics merge into the first or last PW in the long LW.

Comparing LW boundaries obtained by a well developed word segmentation tool with the PW boundaries labeled manually, we found that only 70.7% of the LW boundaries coincided with the PW boundaries, and that 6.4% of the PW boundaries are not LW boundaries. Figure 1 shows the histogram of the lengths of PWs and LWs counted in a large corpus. It can be seen that there are less mono-syllabic PWs than mono-syllabic LWs because most of the

mono-syllabic LWs form disyllabic or tri-syllabic PWs with their neighbors dynamically. Only 1.3% of the PWs contain more than three characters, and the longest PW found in the corpus contains 5 characters. They are often disyllabic or tri-syllabic LWs followed by several clitics, such as "煮熟的了吗？" The higher ratio of disyllabic PWs shows that the PWs reflects the disyllabic rhythm in Mandarin better than the LWs. If speech is synthesized from LWs, the high ratio of mono-syllabic words will decrease the level of naturalness achieved.

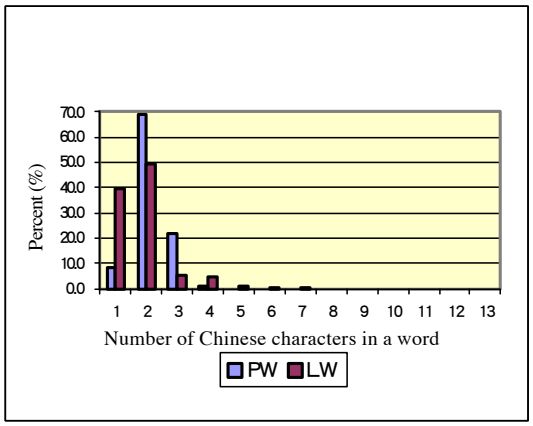


Figure 1 *Histogram of lengths of PWs and LWs in number of characters.*

To investigate the differences between PWs and LWs from the perceptual point of view, a preference experiment was conducted. Speech waveforms were synthesized from two types of input by the MSRCN Mandarin TTS engine [Chu *et al*., 2001]:

A. Sentences were segmented into LW strings, and the LW was used as the basic unit for prosody.
B. Sentences were segmented into PW strings, and the PW was used as the basic unit for prosody.

108 pairs of synthesized speech were played to 15 subjects, who were asked to choose a more natural utterance from each pair. The preference percentages for type A and type B utterances were 21% and 79%, respectively. Speech synthesized from PW strings sounds significantly better than that synthesized from LW strings.

Both the surface difference and perceptual difference between LWs and PWs show that segmenting a sentence into a string of LWs precisely is far from sufficient to generate natural and beautiful prosody in Mandarin TTS systems; it is necessary to re-segment LW strings into PW strings.

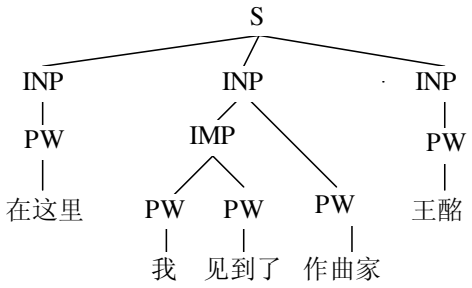
3. Prosodic Constituents in Mandarin

As noted in Section 2, it is very helpful to use the PW instead of the LW as the basic prosody unit. A three-tier instead of the conventional two-tier prosodic hierarchy is defined for a sentence in Mandarin. A sentence consists of one or more INPs. An INP is decomposed into several IMPs and the building blocks for an IMP are PWs. The PW is the lowest constituent in the hierarchy. An INP boundary necessarily coincides with an IMP boundary, and an IMP boundary is inevitably a PW boundary, but the opposite is not true.

Though prosodic constituents should have some relationships with syntactic constituents, the relationships between them are unclear. Figure 2 shows an example sentence "在这里我见到了作曲家王酩 (We saw Wangming, a composer, here)," which is decomposed into a syntactic hierarchy and a prosodic hierarchy. The differences between them are obvious.

A corpus with both prosodic and syntactic labeled structures was prepared. Three-level prosody boundaries were labeled manually after listening to the speech and reading the text transcriptions. Details about the labeling process will be given in Section 5.1. A block-based robust dependency parser [Zhou, 2000] was used to parse all these sentences into syntactic trees. On one hand, only 56.9% of the INP boundaries and 56.4% of the IMP boundaries coincided with the boundaries of top-level syntactic phrases. On the other hand, less than half of the top-level syntactic phrase boundaries were INP boundaries. Figure 3 shows the percentage of syntactic phrase boundaries that coincided with INP boundaries for 7 major syntactic phrase tags. Since great mismatching exists between prosodic phrases and syntactic phrases, directly mapping syntactic phrases to prosodic phrases will cause many unsuitable breaks in synthesized speech. Section 4 will present three approaches to locating prosodic boundaries.

(a)



(b)

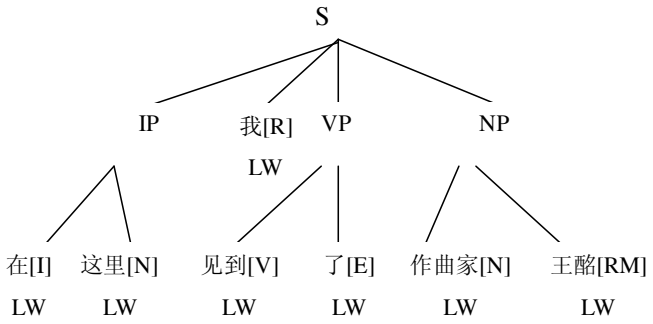


Figure 2 *(a) The prosodic hierarchy and (b) the syntactic hierarchy for the sentence, "*在这里我见到了作曲家王酩 *(We saw Wangming, a composer, here)."*

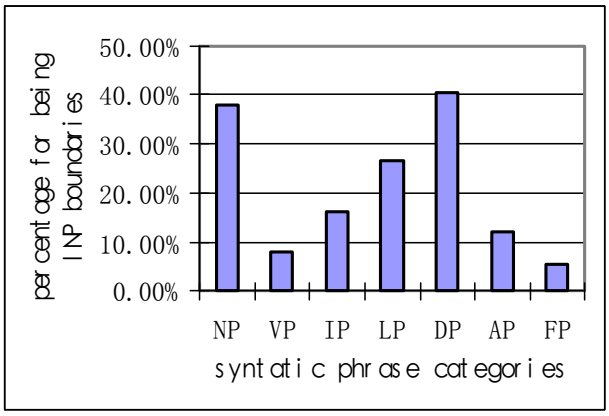


Figure 3 *The percentage of syntactic phrase boundaries that coincided with INP boundaries for 7 major syntactic phrase types. NP - Noun phrase; VP - Verb phrase; IP - Preposition phrase; LP - Post-position phrase; DP - Frame structure; AP - Adjective phrase; FP - Adverb phrase.*

4. Approaches to Locating Prosodic Boundaries

Though, representing prosodic structures by means of syntactic structures directly cannot produce very natural prosody, syntactic information is still helpful for detecting prosodic boundaries. POS has been used in many studies on prosodic phrase prediction. Veilleux *et al*. [Veilleux *et al*., 1990] modeled prosodic group labels and phrase breaks as a six-state Markov chain. Both first- and second- order Markov models were investigated. They reported that using the second-order model did not improve the results. Taylor and Black [Taylor and Black, 1998] used the Markov model in a different way. In their model, state observation probabilities were estimated using a POS sequence model, and the state transition probabilities were estimated using a phrase break model. Wang and Hirschberg [Wang and Hirschberg, 1991] used CART to predict INP boundaries. In Ostendorf and Veilleux' s study [Ostendorf and Veilleux, 1994], CART was used to determine the probability of the occurrence of a minor break at some locations, and a hierarchical stochastic model was used to find the prosodic parse with the highest probability. The Markov model based approaches are based on the assumptions that the current break index is only related to previous indices, and that the state probability and transition probability can be estimated from POS tags of the word sequences. It is difficult to use other syntactic information and length information of phrases and sentences in them. CART based approaches were used in our studies because they can handle data samples with high dimensions, mixed data types and nonstandard data structures. CART based methods also have

the advantage of being comprehensible in the prediction phase. Three predicting models will be presented in this section, and two sets of features will be applied in the training of CART.

Since many Chinese sentences do not have exclusive solutions for LW segmentation and it is possible to have breaks inside some long LWs, each character in a text is assumed to be followed a *potential boundary site* (PBS). Four break indices (BI) are used to label the types of PBS. BI0 represents a non-boundary site. If a PBS is only a PW boundary, it is labeled BI1. BI2 represents an IMP boundary, and BI3 represents an INP boundary. The problem of locating boundaries of prosodic constituents is then changed to the problem of predicting BI for each PBS.

4.1 The basic CART method

CART is used to predict BI for each PBS first. In early CART based approaches [Wang and Hirschberg, 1991; Ostendorf and Veilleux, 1994], features that took continuous values or many discrete values were classified into a limited number of categories first to prevent the excessively dense trees. In many cases, this was done by experts according to their experiences. The number of categories and the way of doing classification would affect the final results. In our approach, the composite-question construction technique [Huang *et al*., 2001] is used to generate complex questions for the tree. The construction of composite questions not only enables flexible clustering of discrete variables, but also produces complex rectangular partitions for continuous variables. Thus, only simple questions about the details of all the features are presented for growing the tree in the training phase.

4.2 The bottom-up hierarchical approach

In the basic CART method, the four BI are treated as being the same, although they have hierarchical relationships. Error analyses show that, sometimes, a BI3 or BI2 is assigned to a non-boundary PBS. This kind of error will decrease not only the naturalness, but also the intelligibility of the synthesized speech. Since PW boundaries can be predicted from LW boundaries with pretty high accuracy [Qian *et al*., 2001], a bottom-up hierarchical approach was proposed. In the new approach, PW boundaries are first detected from all PBS. Then, IMP boundaries are detected only from PBS that are judged to be PW boundaries. Finally, INP boundaries are picked up only from the predicted IMP boundaries. Figure 4 shows the flowchart of the hierarchical approach. Three CARTs were trained separately to make boundary or non-boundary decisions for PWs, IMPs and INPs, respectively. The training procedures for the three CARTs were the same as that described in Section 4.1. However, the data used for training were different. To train the PW-CART, all the PBS with BI0 were treated as non-boundary

samples and all the others were boundary samples. To train the IMP-CART, only PBS with BI1 were used as non-boundary samples, and those with BI2 and BI3 were boundary samples. To train INP-CART, only PBS with BI2 were used as non-boundary samples, and PBS with BI3 were boundary samples.

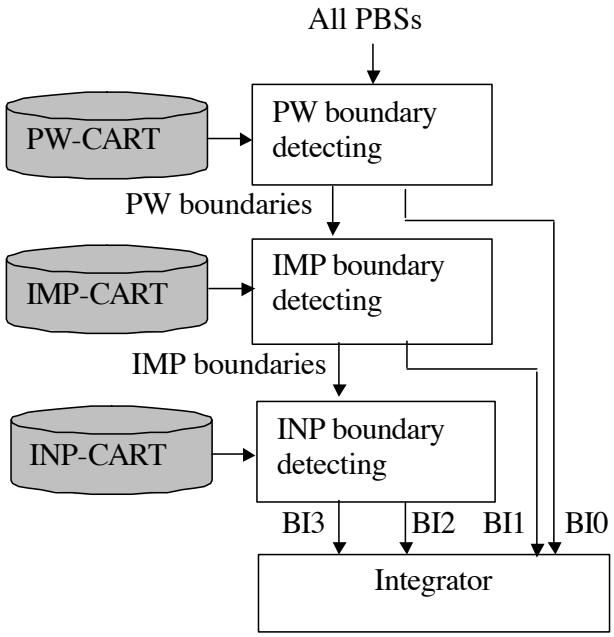


***Figure** 4 Flowchart of the bottom-up hierarchical approach for detecting boundaries of prosody constituents.*

***Table 2.** The average length (ALC) of PWs IMPs and INPs, and their correlation coefficients (CCO) with the lengths of their carrying sentences.*

	PW	IMP	INP
ALC	2.2	3.3	6.7
CCO	0.059	0.155	0.488

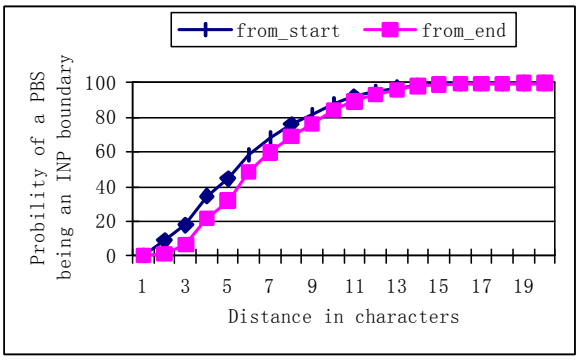


Figure 5 *The probability of a PBS being an INP boundary in terms of its distance to the beginning and the end of a sentence.*

4.3 The modified hierachical approach

In the above two approaches, INP boundaries and IMP boundaries are often confused. We have found that the lengths of sentences in characters, a very important factor that affects the positions of INP boundaries, has not been used sufficiently. The correlation coefficients between the lengths of PWs, IMPs and INPs and the lengths of their carrying sentences, and the average lengths of the three prosodic constituents are listed in Table 2. It can be seen that the length of a PW is uncorrelated with the length of its carrying sentence. The length of an IMP is weakly correlated with the length of its carrying sentence, and the length of an INP is positively correlated with the length of its carrying sentence and tends to increase with it. Statistical results show that the location of an INP boundary is not only related to the length of its carrying sentence, but is related to its distance to the beginning and the end of the sentence. Figure 5 shows two curves revealing the relationship between the probability of a PBS being an INP boundary and its distance to the beginning and the end of its carrying sentence. It is obvious that the PBS at the middle part of a sentence has a higher probability of being an INP boundary than those at the beginning or the end of the sentence. A modified hierarchical approach is proposed based on this observation and another assumption that finding the most likely location for INP boundaries in a sentence one by one is more accurate than finding all INP the boundaries in one loop. In the modified approach, the PW and IMP detecting procedures are the same as those described in Section 4.2. However, the INP detecting procedure is modified to be a recursive detecting method. The output of INP-CART is no longer a boundary or non-boundary decision. Instead, a probability of a PBS being an INP boundary (denoted as P_B) is generated. The use of

INP-CART is similar to that in Ostendorf and Veilleux' s works. P_B for each leaf of the CART is calculated during the training phase. It is defined as the number o f boundary samples over the number of total samples in the leaf. In the prediction phase, when a leaf is selected for an input PBS, its P_B is output as the probability of the PBS being an INP boundary. A confidence measure (denoted as ConfM) for a PBS being an INP boundary is defined by equation (1):

$$ConfM = P_B * P_{start} * P_{end}, \quad (1)$$

where P_{start} and P_{end} are the probabilities of the PBS being an INP boundary in terms of its relative distance to the beginning and the end of the sentence. Their values are defined by the two curves shown in Figure 5, when the distance is smaller than 20. Otherwise they are equal to 1.

The recursive INP boundary detecting algorithm is decomposed into four steps.

Step1: ConfM values are calculated for all PBS that have been detected as IMP boundaries by the IMP-CART. BI3 is assigned to the one with the highest ConfM value, if its ConfM value is larger than the pre-set threshold θ_{ConfM} . If no PBS with a ConfM value larger than θ_{ConfM} is found, go to Step 4.

Step2: Split the sentence into two parts at the found INP boundary.

Step3: Repeat Step1 and Step 2 for the two new sub-sentences recursively until all paths reach Step 4.

Step4: Stop.

The performance changes with the value of θ_{ConfM} , and it is set according to previous experience or experiments. In our case, the best result was achieved when $\theta_{ConfM} = 0.105$.

5. Experiments

Experiments using the three methods described in Section 4 were carried out on a large speech corpus. The speech corpus, features used, and results from the experiments will be discussed in this section.

5.1 Speech corpus

Since there was no public Mandarin speech database available for this study, we designed and collected a large phonetically and prosodically enriched Mandarin speech corpus. The corpus contains about 12,000 utterances (sentences), which were uttered by a professional female speaker. Prosodic indices BI1 to BI3 were annotated manually by listening to these utterances and reading the text transcriptions. BI1 was annotated according to the guidelines listed in Table 1. BI2 and BI3 were labeled according to the breaks heard. When a minor break was perceived,

a BI2 was assigned. When a major break was heard, a BI3 was assigned. BI2 and BI3 were assumed to correspond to IMP and INP boundaries, respectively. The end of each utterance was labeled with BI3, and each non-boundary PBS was labeled with BI0 automatically.

To check consistency of annotation across different people, an exploratory experiment was carried out. Three annotators were first trained on the same 100 sentences. At this stage, they were required to discuss criteria for annotation so that they could achieve agreement on most of the annotations in the 100 sentences. Then, they were asked to annotate a small subset of the corpus, which included 1,348 sentences and 1,8983 PBS. All three annotators achieved agreement on 82.9% of BIs, and 99.1% of BIs were agreed to by at least two of them. That is to say pretty good consistency existed among the three annotators. To reduce costs, the whole speech corpus was only annotated by one of them.

Investigating the relationship between BI types and punctuation, such as commas, colons and semicolons, we found that there were altogether 5,718 items of punctuation in the corpus (full stops at the ends of sentences were excluded), 5,693 (99.6%) of which were related to BI3 and the rest related to BI2. These kinds of punctuation are referred to as breaking punctuation (BP). Since BPs almost always imply INP boundaries, no learning process is needed for them. All PBS with BPs were assigned to BI3. This has been done in many Mandarin TTS systems. However, placing major breaks only at PBS with BPs is not adequate for synthesizing high quality speech. The ability to predict INP boundaries at PBS without BPs is more important. Thus, accuracy for INP boundaries is calculated using two constraints in this paper. In one constraint, all predicted INP boundaries, including INP boundaries at PBS with BPs, are considered. In the other constraint, only INP boundaries at PBS without BPs are taken into account.

Most of the early studies on detecting prosodic phrases experimented on small databases. Wang and Hirschberg used a 298-utterance corpus, and Ostendorf and Veilleux used 312 sentences in their experiments. Only a limited number of INP boundaries can be found in such small corpora. Thus, only a few features can be used in the training and testing phase to avoid sparsity of training data. A larger training and testing data set was used in this study. 2,583 sentences with 38,499 characters from the corpus we collected were used for training, and another 1,000 sentences with 15,618 characters were used for testing.

5.2 Feature set

Although both acoustic features [Wightman and Ostendorf, 1994; Chou *et al*., 1998], such as f_0, duration and energy, and syntactic features [Hirschberg and Prieto, 1990; Wang and Hirschberg, 1991; Ostendorf and Veilleux, 1994; Lee and Oh, 1999], such as POS tags and syntactic phrasal

information, have been used to label break indices, only features that can be derived from texts were used in this study. The reason is that no acoustic feature is available when we predict prosodic boundaries in TTS systems. Two feature sets with or without syntactic phrasal information were used. Set 1 is the one without syntactic phrase information. Features used in Set 1 are listed as follows:

1) POS for LWs around each PBS are the most commonly used features in prosodic phrase prediction. A window of three words is used in our approach: two words before and one after the PBS. 26 POS tags are used. Among them, 9 are the normal POS tags used by Zhou's parser [Zhou, 2000]. The others are characters or words that often have special effects on prosodic boundaries. These characters and words are obtained through data analyses and should be considered individually. All 26 tags are listed in Table 3.

Table 3. *The 26 POS tags used in our experiments*

Tags	Explanation	Tags	Explanation	Tags	Explanation
N	Noun	Char1	Mono-syllabic LW "电"	Char 10	Mono-syllabic LW "从"
V	Verb	Char2	Mono-syllabic LW "中"	Word1	Disyllabic LW "但是"
A	Adjective	Char3	Mono-syllabic LW "后"	Word2	Disyllabic LW "目前"
F	Adverb	Char4	Mono-syllabic LW "的"	Word3	Disyllabic LW "今天"
DM	Place name	Char5	Mono-syllabic LW "在"	Word4	Disyllabic LW "短波"
RM	Person name	Char6	Mono-syllabic LW "于"	Word5	Disyllabic LW "简讯"
QM	Organization name	Char7	Mono-syllabic LW "了"	Word6	Disyllabic LW "接着"
E-I-L-J	Auxiliary, preposition, post-preposition and junction	Char8	Mono-syllabic LW "等"	Word7	Disyllabic LW "就是"
Other	All other POS	Char 9	Mono-syllabic LW "着"		

2) The length in characters of the LW in the window is very important for predicting PW boundaries. It takes 5 discrete values: 1- 4 represent LWs containing 1-4 characters,

respectively. 5 represents all LWs containing more than 4 characters.

3) The distance in characters from the current PBS to the beginning or the end of a sentence. The shorter one among the two is used. As shown in Figure 5, the lengths are divided into four groups, which are <=2, 3-6, 7-10 and >10, respectively.
4) The lengths in characters of the carrying sentences are divided into three groups, which are <=10, 11-20 and >20, respectively.

Set 2 contains all the features in set 1 and the phrasal features listed below:

1) Whether the current PBS is a top-level major syntactic phrase boundary or not.
2) The phrase category for the carrying phrase of the current PBS. The 7 categories used by Zhou [Zhou, 2000] are used. The seven phrase categories are NP - Noun phrase; VP - Verb phrase; IP - Prepositional phrase; LP - Post-position phrase; DP - Frame structure; AP - Adjective phrase; FP - Adverb phrase.
3) The length of the carrying phrase of the current PBS. The lengths are divided into five groups, which are <=5, 6-10, 11-15, 16-20 and >20, respectively.

5.3 Evaluation criteria

There is no commonly accepted measure for evaluating the performance of prosodic parsers. Wang and Hirschberg used accuracy. Accuracy reflects the average performance in both breaking and non-breaking cases. However, what we really care about is the performance in breaking cases. Furthermore, the ratio of the number of breaking samples to that of non-breaking samples greatly affects the overall accuracy. For example, 95% and 94% accuracy for English and Spanish were reported by Hirschberg and Prieto. However, from the CART prediction tree for Spanish given in their paper, we find that only about 16.4% of the total samples had breaks. That is to say if all the samples are predicted to be non-breaking, then 83.6% accuracy is still obtained. The same measure was used by Lee and Oh in their experiments on Korean. Only 85% accuracy was reported. We find the reason for the drop in accuracy is that their testing set contained many more breaking samples (37%). Several measures were used together for evaluation in Taylor and Black's study. They were breaks-correct, the ratio of correctly predicted breaks to all real breaks, junctures-correct, which is the same as the accuracy measure used by Wang and Hirschberg, and juncture-insertion, the total number of insertion errors over the number of data. Juncture-insertion is not an efficient measure. In this study, four measures were used together to evaluate performance. Precision and recall were calculated for each BI type separately, and they are defined by equation (2) and (3), respectively:

$$\Pr e_j = Count(B_{cpj}) / Count(B_{pj}) \ , \tag{2}$$

$$\mathrm{Re}\, c_j = Count(B_{cpj}) / Count(B_{rj}) \ , \tag{3}$$

where j = 0, 1, 2 or 3 denotes the type of BI, $Count(B_{pj})$ is the total number of predicted boundaries for BIj, $Count(B_{cpj})$ is the number of BIj that are predicted correctly and $Count(B_r)$ is the number of real BIj.

Overall accuracy for all BI is calculated using equation (4):

$$Accu = \sum_{j=0}^{3} Count(B_{cpj}) / \sum_{j=0}^{3} Count(B_{rj}) \ . \tag{4}$$

In our study, we found that different types of errors would reduce the naturalness of the synthesized speech to different extents. The larger the BI error, which is defined as the difference between the assigned index and the real one, the larger the decrease in quality. Therefore, an overall error cost is defined by equation (5):

$$ErrCost = \sum W_i Count(E_i) \ , \tag{5}$$

where E_i represents the case where the number of BI errors equals *i*. In our case, only three types of errors, E_1, E_2 and E_3, exist. $Count(E_i)$ is the total number of E_i errors, and W_i represents the weight for E_i. In this study, $W_1 = 0.5$, $W_2 = 1$ and $W_3 = 2$.

5.4 Results

5.4.1 Basic CART method

CART was trained with both feature sets over the same training set. Only simple questions about each individual category of each feature in the feature set were provided manually. Composite questions were constructed automatically. A composite question was formed by first growing the tree with several simple questions and then clustering the leaves into two sets [Huang *et al*., 2001]. Multiple OR and AND were used to form a composite question for each set. In our case, the depth for search a composite question was five split. The growing of the tree stopped when 40 composite questions had been formed. We have compared the results from 20, 40 and 60 composite questions. 40 was better than 20 in most cases. However, 60 was not better than 40. Thus, 40 composite questions were used in all the training phases for CART in this study.

The four measures obtained by testing the CARTs growing from the two feature sets are listed in Table 4. Column BI3NP shows the precision and recall for BI3 at PBS without BPs. The precision and recall for BI3 in this column is more meaningful than that in column BI3. According to Table 4, feature set 2 produced 1% increase in overall accuracy and 11% decrease in the error cost, compared to set 1. Table 4 also shows that syntactic phrasal information

benefited the precision and recall results for BI2 and BI3 more. However, this improvement was achieved at the cost of using a syntactic parser in on-line systems. Furthermore, the online syntactic parse cannot always provide reliable phrasal information. The phrasal information used in this study was checked manually. If the tags generated from the syntactic parser had been used directly, much worse results would have been obtained. Thus, only feature set 1 was used in the experiments with the other two approaches.

Table 4. *The performance of prosodic boundary prediction with the basic CART method for the two feature sets.*

Feature set	Evaluation Criteria	BI0	BI1	BI2	BI3	BI3NP
Set 1	precision(%)	93.19	63.95	57.41	81.12	65.66
	recall(%)	95.92	69.1	55.68	59.43	39.47
	overall accuracy (%)	82.48				
	overall error-cost	1694.5				
Set 2	precision(%)	95.01	65.06	57.77	83.67	69.85
	recall(%)	95.98	66.13	64.18	60.22	40.64
	overall accuracy (%)	83.41				
	overall error-cost	1508				

5.4.2 Bottom-up hierarchical method

The three CARTs shown in Figure 4 were trained separately from the same training set. Only feature set 1 was used. The precision and recall results for each individual CART are listed in Table 5. The integrated results are listed in Table 6. A significant decrease in the precision and recall performance for BI1, BI2 and BI3 were observed when the outputs from the three CARTs were integrated. The reason may be that errors from PW-CART were promulgated into IMP- and INP-CART, and errors from IMP-CART were promulgated into INP-CART. Comparing Table 6 and Table 4, the same overall accuracy was obtained on feature set 1. However, a 7.2% reduction in the error-cost was achieved, which means that errors with larger BI differences were reduced.

Table 5. *The performance of each individual CART.*

	PW-CART	IMP-CART	INP-CART
Precision (%)	95.74	80.96	84.77
Recall (%)	96.15	87.68	64.90

Table 6. *The integrated results for the bottom-up hierarchical method.*

Feature set	Evaluation Criteria	BI0	BI1	BI2	BI3	BI3NP
Set 1	precision(%)	95.30	65.61	53.27	81.44	67.48
	recall(%)	95.73	58.57	65.61	62.58	44.17
	overall accuracy(%)	82.49				
	overall error-cost	1590				

5.4.3 Modified hierarchical method

The three CARTs trained as described in Section 5.4.2 were also used in this modified version. BI1 and BI2 were predicted step by step as described in the previous section. However, INP boundaries were predicted using the recursive method described in Section 4.3. The final results are listed in Table 7. Comparing Table 7 with Table 6, the precision and recall performance for BI0 and BI1 are unchanged and that for BI2 and BI3 are improved. A 0.6% increase in overall accuracy and a 5.6% reduction in the error-cost are observed. The best precision and recall performance was obtained for BI3 at PBS without BP. All these improvements show that the recursive prediction method benefits the prediction of BI3.

Table 7. *The performance of BI assignment at PBS using the modified hierarchical approach.*

Feature set	Evaluation Criteria	BI0	BI1	BI2	BI3	BI3NP
Set 1	precision(%)	95.30	65.61	54.70	82.68	71.12
	recall(%)	95.73	58.57	65.61	68.10	52.41
	overall accuracy (%)	82.99				
	Overall error-cost	1550.5				

5.5 Experiment on acceptability

While manually annotated break indices are used as a reference for evaluating the results obtained using automatic methods, they are not the only correct indices since the same sentence can be spoken in different ways by human. Two experiments were conducted to evaluate the acceptability of the mis-assigned BI.

5.5.1 Experiment 1

All the errors generated by the modified hierarchical method were presented to three subjects. If at least two of them thought that the mis-assigned break index was acceptable, then, it was considered as a felicitous error. Otherwise, it was considered as an infelicitous error. Among the 2,657 errors, only 698 (26.3%) were infelicitous.

5.5.2 Experiment 2

100 sentences in the testing set were used in this experiment. Two sets of waveforms were synthesized using a data-driven TTS system [Chu *et al*., 2001]. Set A was synthesized from the scripts with manually annotated break indices, and Set B was generated from the scripts with the automatically labeled break indices. The two versions of synthetic waveforms of one sentence formed two pairs of stimuli in the sequence AB, BA. The 200 stimuli were played to 12 subjects, who had to select one from each pair that sounded more natural. The preference rate was calculated as $P_j = count(T_j) / \sum count(T_j)$, (6)

where $count(T_j)$ is the total number of times type T_j is preferred; j=A or B.

The preference rates for the two sets of synthetic sounds are shown in Figure 6. It can be seen that P_A was higher than P_B, but that the difference between them was not very large. This result shows that our automatic method generated rather natural break indices, which were acceptable in most cases.

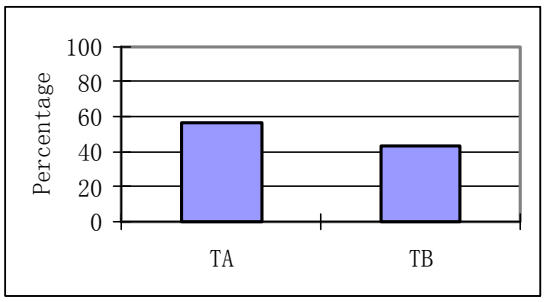


Figure 6 *Preference rates for the two types of synthesized speech. TA, speech synthesized from scripts with manually annotated BI; TB, speech synthesized from scripts with automatically generated BI.*

5.6 Discussion

Three approaches have been proposed in this section for locating the three-tier prosodic boundaries in unrestricted Mandarin texts. Because of differences in language, training and testing corpora, and the definition of prosodic constituent to be predicted, comparing results obtained in different experiments is not easy. The overall accuracy (83%) achieved in our study is not as high as that reported by Hirschberg and Prieto (95%), Lee and Oh (85%). However,

their experiments only involved making decisions between breaks and non-breaks. However, three levels of prosodic boundaries were detected in this study. Another reason for the drop in overall accuracy is the difference in the ratio of the number of break samples to that of non-break samples. In Hirschberg and Prieto' s experiment, about 16.4% of the total samples were breaks. That is to say, if all the testing data are assigned a non-break index, then 83.6% accuracy can still be obtained. In Lee and Oh' s experiment in Korean, the 37% break samples caused a significant drop in accuracy. In our testing data, only 54% were non-boundary samples. Thus, the 83% overall accuracy for the four BI is not poor performance. In all the previous studies, punctuation was used as a very important feature. However, we found that a piece of breaking punctuation almost always implied an INP boundary. Thus, predicting boundaries from non-punctuation PBS should be the focus of studies on locating boundaries. The precision and recall results obtained in several studies on BI3 at PBS with or without BP are listed in Table 8. We derived these results from the tables listed in their papers. From Table 8, the advantages of our method for PBS without BP are obvious.

Table 8. *A comparison of the performance achieved in predicting major breaks with previous results. A "+" means that the corresponding number can not be derived from the original paper.*

Comparing condition	Evaluation Criteria	Hirschberg and Prieto	Lee and Oh	Taylor and Black	Ours
BI3	Precision	92.3%	77.1%	72.3%	82.68%
	Recall	72.4%	85.4%	79.3%	68.10%
BI3NP	Precision	72.1%	+	49.3%	71.12%
	Recall	31.5%	+	54.7%	52.41%

6. Conclusion

This paper has proposed a three-tier prosodic hierarchy, which emphasizes the use of the PW instead of the LW as the basic prosodic unit. Both the surface difference and perceptual difference show the advantages of this prosodic hierarchy. Three approaches to locate the boundaries of prosody constituents in unrestricted Mandarin texts have been presented. The syntactic phrasal information produced a 1% increase in accuracy and an 11% decrease in the error cost for the basic CART method. The improved hierarchical method achieved the best performance on feature set 1. It also produced the best performance in finding INP boundaries. The two acceptability experiments revealed that only 26.3% of the mis-assigned break indices were actually infelicitous errors, and that the perceptual difference between the automatically assigned break indices and the manually annotated break indices was not large.

In this study, modified hierarchical approach, INP-CART was used to generate the probability of each PBS being a boundary. It may not be the best algorithm for generating this probability. A better algorithm may be found in our future work.

Acknowledgements

The authors thank Dr. Ming Zhou for providing the block-based robust dependency parser as a toolkit for use in this study. Thanks go to everybody who took part in the perceptual test. The authors are especially grateful to all the reviewers for their valuable remarks and suggestions.

References

Chou F. C., Tseng, C.Y. and Lee, L.S., "Automatic generation of prosodic structure for high quality Mandarin speech synthesis", *Proceeding of the Fourth International Conference on Spoken Language Processing*, 1996, Philadelphia.

Chou F. C., Tseng, C.Y. and Lee, L.S., "Automatic segmental and prosodic labeling of Mandarin speech database", *Proceeding of the Fifth International Conference on Spoken Language Processing*, 1998, Sydney.

Chu, M, Peng, H., Yang, H. and Chang, E. "Selection non-uniform units from a very large corpus for concatenative speech synthesizer", *Proceeding of the 2001 International Conference on Acoustics, Speech and Signal Processing*, 2001, Salt Lake City.

Chu, M. and Lu, S. N., "A Text-to-speech System with High Intelligibility and High Naturalness for Chinese", *Chinese Journal of Acoustics*, Vol.15, No.1, 1996, pp. 81-90.

Dutoit, T., Pagel, V., Pierret, N., Bataille, F. and Verchen, O., "The MBTOLA Project: Towards a set of high quality speech synthesizes free of use for no commercial purpose", *Proceeding of the Fourth International Conference on Spoken Language Processing*, 1996, Philadelphia.

Gee, J. P. and Grosjean, F., "Performance Structure: A Psycholinguistic and Linguistic Appraisal", *Cognitive Psychology*, Vol. 15, 1983, pp. 411-458.

Hirschberg, J. and Prieto, P., "Training intonatinal phrasing rules automatically for English and Spanish text-to-speech", *Speech Communication*, Vol. 18, 1996, pp. 281-290.

Huang, X. D., Acero, A. and Hon, H. W., "Chapter 4: Pattern Recognition", *Spoken Language Processing – A Guide to Theory, Algorithm and System Development*, 2001, Prentice Hall PTR.

Ladd, D. R. and Campbell, N., "Theories of Prosodic Structure: Evidence from Syllable Duration", *Proceeding of the 12nd International Congress of Phonetic Sciences*, 1991.

Lee, S. and Oh, Y. H., "Tree-based modeling of prosodic phrasing and segmental duration for Korean TTS systems", *Speech Communication*, Vol. 28, 1999, pp. 283-300.

Liberman, M. Y. and Prince, A. S., "On Stress and Linguistic Rhythm", *Linguistic Inquiry*, Vol. 8, 1977, pp. 249-336.

Ostendorf, M. and Veilleux, N., "A hierarchical stochastic model for automatic prediction of prosodic boundary location", *Computational Linguistics*, Vol.20, No.1, 1994, pp. 27-54.

Qian, Y., Chu, M. and Peng, H., 2001. "Segmenting unrestricted Chinese text into prosodic words instead of lexical words", *Proceeding of the 2001 International Conference on Acoustics, Speech and Signal Processing*, 2001, Salt Lake City.

Selkirk, E., *Phonology and syntax: The relationship between sound and structure*, MIT press, 1984.

Shen, X. and Xu, B., "A CART based hierarchical stochastic model for prosodic phrasing in Chinese", *Proc. of the 2nd International Symposium on Chinese Language Processing*, 2000, Beijing.

Taylor, P. and Black, A.W., "Assigning phrase breaks from part-of-speech sequences", *Computer speech and language*, Vol. 12, 1998, pp. 99-117.

Veilleux, N.M., Ostendorf, M., Price, P.J. and Shattuck-Hufnagel, S., "Markov Modeling of Prosodic Phrase Structure", *Proceeding of the 1990 International Conference on Acoustics, Speech and Signal Processing*, Vol.2, 1990, pp. 777-780.

Wang, M.Q. and Hirschberg, J., "Predicting intonational phrasing from text", *Proceeding of Association for Computational Linguistics 29th annual meeting*, 1991, pp. 285-292.

Wightman, C.W. and Ostendorf, M., "Automatic labeling of prosodic patterns", *IEEE Trans. on Speech and Audio Processing*, Vol.2, No.4, 1994, pp. 469-481.

Zhou, M., "A block-based robust dependency parser for unrestricted Chinese text", *Proceeding of the second Chinese Language Processing Workshop Attached to ACL2000*, 2000, Hong Kong.

Computational Linguistics and Chinese Language Processing
Vol. 6, No. 1, February 2001, pp. 83-108

Automatic Translation Template Acquisition Based on Bilingual Structure Alignment[1]

Yajuan Lü[*], Ming Zhou[+], Sheng Li[*],

Changning Huang[+], Tiejun Zhao[*]

Abstract

Knowledge acquisition is a bottleneck in machine translation and many NLP tasks. A method for automatically acquiring translation templates from bilingual corpora is proposed in this paper. Bilingual sentence pairs are first aligned in syntactic structure by combining a language parsing with a statistical bilingual language model. The alignment results are used to extract translation templates which turn out to be very useful in real machine translation.

Keywords: Bilingual corpus, Translation template acquisition, Structure alignment, Machine translation

1. Introduction

Bilingual corpora have been recognized as a valuable resource for knowledge acquisition in machine translation and many other NLP tasks. To make better use of them, bilingual corpora are often aligned first. Intensive researches have been done on sentence and word level alignment [Brown *et al*. 1991, Church 1993, Ker *et al*. 1997, Huang *et al*. 2000]. These alignments have been proven to be very useful in machine translation, word sense disambiguation, information retrieval, translation lexicon extraction, and so on. With a sentence aligned parallel English-Chinese corpus ready in hand, this paper extends word-level alignment to syntactic structure alignment with the aim of acquiring structural translation templates automatically.

---

[1] The work was partially done at Microsoft Research Asia.

[*] Dept. of Computer Science & Engineering, Harbin Institute of Technology, Harbin, 15001
E-mail: Lü Yajuan: lyj@mtlab.hit.edu.cn Zhao Tiejun: tjzhao@mtlab.hit.edu.cn

[+] Microsoft Research Asia, Beijing, 10080
E-mail: Zhou Ming: mingzhou@microsoft.com Huang Changning: cnhuang@microsoft.com

Numerous researches have been done to acquire knowledge from bilingual corpora. Many of these studies aimed to acquire word or phrase translation lexicons [Shin *et al*. 1996, Fung *et al*. 1997, Ralf 1997, Turcato 1998]. This paper focuses on the automatic learning of translation templates, which are especially useful for machine translation. In [Guvenir *et al*. 1998], [Malavazos *et al*. 2000] and [Cicekli *et al*. 2001], analogical models were proposed to learn translation templates. By grouping similar translation examples and replacing their difference with a variable, they could obtain translation templates. Structure alignment has been studied by several researchers for use in structural translation template acquisition. Most of the approaches have followed what may be called a "parse-parse-match" procedure [Wu 1997]. The main idea is that each language of the parallel corpus is first parsed individually using a monolingual grammar, and then the corresponding constituents are matched using some heuristic procedures. The works by [Kaji *et al*. 1992], [Almuallim *et al*. 1994], [Grishman *et al*. 1994], [Matsumoto *et al*. 1995], [Meyers *et al*. 1998], [Watanabe *et al*. 2000] etc. can be considered such approaches. Differences between them are in their parsing grammars and heuristic procedures. Kaji and Watanabe used phrase structure grammar, while Grishman employed a regularized syntactic structure. The dependency structure is used in most of the other systems. In [Watanabe 1993], bilingual structure matching was used to improve the existing transfer rules by comparing in incorrect translation and correct translation. Wu [Wu 1995a, Wu 1997] proposed a bilingual language model to represent a bilingual corpus and parse bilingual sentences simultaneously. Because of the lack of a suitable bilingual grammar, their system is used to acquire phrase translation examples, not templates. In all these studies, structure-aligned bilingual corpora were shown to be very useful for translation knowledge acquisition.

The method proposed in this paper differs from the previous approaches in two ways: (1) The bilingual structure alignment is based on a bilingual language model and uses only one language parsing result. Compared with the "parse-parse-match" procedure, monolingual parsing is particularly suitable when there is no robust parser for one of the languages (such as Chinese). (2) The translation templates we acquire are integrated with the processes of transfer and generation, which are the usual two phases in machine translation systems. Two types of templates are obtained: structure translation templates and word selection templates.

This paper is organized as follows: In the next section, we propose a bilingual structure alignment algorithm by combining a language parsing with a statistical bilingual language model. Then, the learning of translation templates is described in section 3. A translation experiment based on the acquired knowledge is described in section 4. We conclude our work in section 5. Although this paper is related to English-Chinese structure alignment and template acquisition, the proposed method is also applicable to other language pairs because it

is language independent.

2. Bilingual structure alignment using monolingual parsing

The "parse-parse-match" procedure for bilingual structure alignment is susceptible to three weaknesses: [Wu 1995a]

- Appropriate, robust, monolingual grammars may not be available for both languages. This is the case when Chinese is one of the languages.
- The parsing grammars used in the two languages may be incompatible.
- The process of selecting between multiple possible arrangements may be arbitrary.

To overcome these weaknesses, Wu [Wu 1995d, Wu 1997] has proposed a bilingual language model called the Inversion Transduction Grammar (ITG), which can be used to parse bilingual sentence pairs simultaneously. Subsection 2-1 will give a brief description. For details please refer to [Wu 1995a, Wu 1995b, Wu 1995c, Wu 1995d, Wu 1997]. Based on this model, a bilingual structure alignment algorithm guided by one language parsing will be presented in subsection 2-2.

2.1 ITG bilingual language model

The Inversion Transduction Grammar is a bilingual context-free grammar that generates two matched output languages (referred to as L_1 and L_2). It also differs from standard context-free grammars in that the ITG allows right-hand side production in two directions: straight or inverted. The following examples are two ITG productions:

$$C \rightarrow [A\ B],$$
$$C \rightarrow \langle A\ B \rangle.$$

In the above productions, each nonterminal symbol stands for a pair of matched strings. For example, the nonterminal A stands for the string-pair (A_1, A_2). A_1 is a sub-string in L_1, and A_2 is A_1's corresponding translation in L_2. Similarly, (B_1, B_2) denotes the string-pair generated by B. The operator [ ] performs the usual concatenation, so that $C \rightarrow [A\ B]$ yields the string-pair (C_1, C_2), where $C_1=A_1B_1$ and $C_2=A_2B_2$. On the other hand, the operator <> performs the straight concatenation for language 1 but the reversing concatenation for language 2, so that $C \rightarrow \langle A\ B \rangle$ yields $C_1=A_1B_1$, but $C_2=B_2A_2$. The inverted concatenation operator permits the extra flexibility needed to accommodate many kinds of word-order variation between source and target languages [Wu 1995b].

There are also lexical productions of the following form in ITG:

$$A \rightarrow x/y,$$

which means that a symbol x in language L_1 is translated by the symbol y in language L_2. The x, y may be a null symbol e, which means there may be no counterpart string in the other language.

Parsing, in the case of an ITG, means building matched constituents for an input sentence-pair. For example, Figure 1 shows an ITG parsing tree for an English-Chinese sentence-pair. The inverted production is indicated by a horizontal line in the parsing tree. The English text is read in the usual depth-first left to right order, but for the Chinese text, a horizontal line means the right sub-tree is traversed before the left. The generated parsing results are:

(1) a. $[[[\text{The game}]_{BNP}\ [[\text{will start}\ ]_{VBP}\ [\text{on Wednesday}]_{PP}\ ]_{VP}\ ]_S\ .]_S$

b. $[[\text{比赛}\ [\text{星期三}\ \ \text{开始}]_{VP}\ ]_S\ \circ\ ]_S$

We can also represent the common structure of the two sentences more clearly and compactly with the aid of <> notation:

(2) $[[[\text{The/e game/比赛}]_{BNP} < [\text{will/e start/开始}]_{VBP}\ [\text{on/e Wednesday/星期三}\ \ ]_{PP} >_{VP}\ ]_S\ ./\circ\ ]_S$

where the horizontal line from Figure 1 corresponds to the <> level of bracketing.

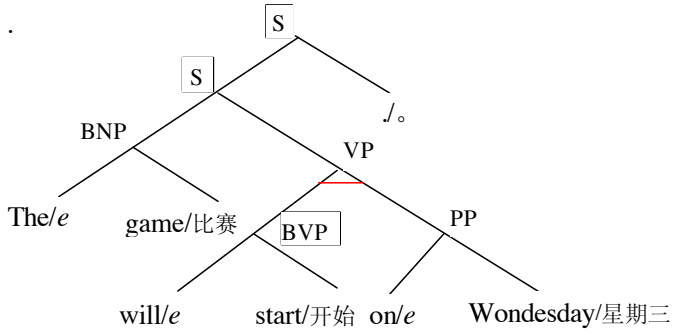


Figure 1 *Inversion transduction grammar parsing tree.*

Any ITG can be converted to a normal form, where all productions are either lexical productions or binary-fanout nonterminal productions [Wu 1995b, Wu 1995c, Wu 1997]. If probability is associated with each production, the ITG is called the Stochastic Inversion Transduction Grammar (SITG).

Because of the difficulty of finding a suitable bilingual syntactic grammar, a practical ITG is a generic Bracketing Inversion Transduction Grammar (BTG), which has been used by Wu in several experiments on bilingual bracketing and to extract phrasal translation examples

[Wu 1995a, Wu 1995b, Wu 1995c]. BTG is a simplified ITG that has only one nonterminal and does not use any syntactic grammar. A Statistical BTG (SBTG) grammar is as follows:

$$A \xrightarrow{a} [AA]; \quad A \xrightarrow{a} < AA >; \quad A \xrightarrow{b_{ij}} u_i / v_j ; \quad A \xrightarrow{b_{ie}} u_i / e; \quad A \xrightarrow{b_{ej}} e / v_j .$$

SBTG employs only one nonterminal symbol *A* that can be used recursively. Here, "*a*" denotes the probability of syntactic rules. However, since those constituent categories are not differentiated in BTG, it has no practical effect here and can be set to an arbitrary constant. The remaining productions are all lexical. b_{ij} is the translation probability that source word u_i translates into target word v_j. b_{ij} can be obtained using a statistical word-translation lexicon[Wu 1997] or statistical word alignment[Lü *et al*. 2001]. The last two productions denote that the word in one language has no counterpart in another language. A small constant can be chosen for the probabilities b_{ie} and b_{ej}.

In BTG, no language specific syntactic grammar is used. The maximum-likelihood parser selects the parse tree that best satisfies the combined lexical translation preferences, as expressed by the b_{ij} probabilities. Because the expressiveness characteristics of ITG naturally constrain the space of possible matching in a highly appropriate fashion, BTG achieves encouraging results for bilingual bracketing using a word-translation lexicon alone [Wu 1995a].

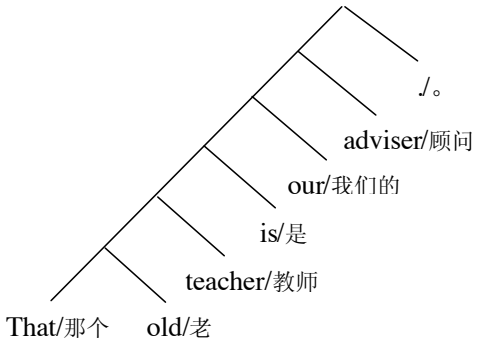
Since no syntactic knowledge is used in SBTG, output grammaticality can not be well guaranteed. In particular, if the corresponding constituents appear in the same order in both languages, both straight and inverted, then lexical matching does not provide the discriminative leverage needed to identify the sub-constituent boundaries. For example, consider an English-Chinese sentence pair:

(3) English: That old teacher is our adviser.
Chinese: 那个老教师是我们的顾问。

The SBTG parsing tree is shown in Figure 2(a), and the corresponding bracketing result is shown in Figure 2(b). The result does not accord with the syntactic structure as we expected. In this case, grammatical information about one or both of the languages can be very helpful. For example, if we know the English parsing result shown in (a) in Figure 3, then the bilingual parsing can be determined easily; the result should be that shown in (b), and the corresponding bracketing result is that shown in (c).

.

(a)

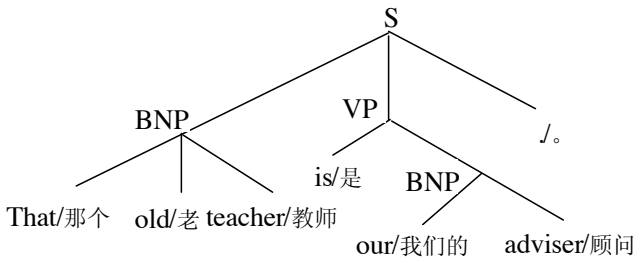


(b) [[[[[[The/那个 old/老] teacher/教师] is/是] our/我们的] adviser/顾问] ./。]

Figure 2 *Bilingual parsing with SBTG.*

(a) English parsing: $[[\text{That old teacher}]_{\text{BNP}}\ [\text{is}\ [\text{our adviser}]_{\text{BNP}}\ ]_{\text{VP}}\ .]_{\text{S}}$

(b)



(c) $[[\text{That/那个 old/老 teacher/教师}]_{\text{BNP}}\ [\text{is/是}\ [\text{our/我们的 adviser/顾问}]_{\text{BNP}}\ ]_{\text{VP}}\ ./。\ ]_{\text{S}}$

Figure 3 *Bilingual parsing guided by English parsing.*

Statistics in a corpus of 20,000 word-aligned sentence-pairs indicates that nearly 72% of the sentence-pairs contain the corresponding constituents, which include more than three continuous sub-constituents in identical order. These constituents often lead to ungrammatical parsing with SBTG. Therefore, it is necessary to introduce a language grammar in ITG instead of not using any grammar as in BTG.

2.2 Integrating monolingual parsing with a bilingual language model

From the above discussion, we can see that if one language parser is available, then the bilingual bracketing result can be more grammatical. This is important for syntactic translation template acquisition.

English parsing methods have been well studied. We have also developed an incremental English parser using statistic and learning methods [Meng *et al*. 2001]. A structure alignment algorithm guided by English parsing will be described in this section.

Here, structure alignment guided by English parsing means using an English parser's bracketing information as a boundary restriction in the ITG language model. But this does not necessarily mean parsing the other language completely according to the same parsing boundary. If a parsing structure is fixed according to one language, it is possible that the structure is not linguistically valid for the other language under the formalism of Inversion Transduction Grammar. To illustrate this, see the example shown in Figure 4.

The sub-trees for each blacked underlined part are shown in Figure 4(a) and (b). We can see that the Chinese constituents do not match the English counterparts in the English structure. In this case, our solution is that shown in Figure 4 (c): the whole English constituent of "VP" is aligned with the whole Chinese correspondence; i.e., "eat less bread" is matched with "**少吃面包**." At the same time, we give the inner structure matching according to SITG regardless of the English parsing constraint. An "X" tag is used to indicate that the sub-bilingual-parsing-tree is not consistent with the given English sub-tree.

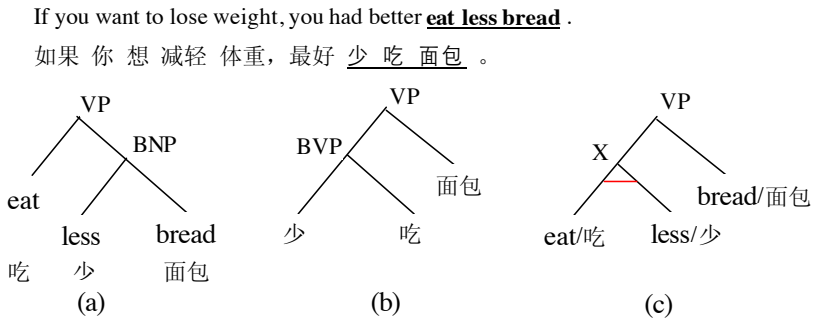


Figure 4 *An example of mismatched sub-trees.*

The main idea is that the given parser is only used as an boundary constraint for bilingual parsing. When the constraint is incompatible with the bilingual model ITG, we use ITG as the default result. This process enables parsing go on regardless of some failures in matching.

We heuristically define a constraint function $F_e(s, t)$ to denote the boundary constraint, where s is the beginning position and t is the end. There are three cases of structure matching: *violate match, exact match* and *inside match*. *Violate match* means the bilingual parsing conflicts with the given English bracketing boundary. (1,2), (1,3), (2,3), (2,5) etc. in the following English sentence (4) are examples. We assign a minimum $F_e(s, t)$ (0.0001 at present) to prevent the structure match from being chosen when an alternative match is available. *Exact match* means the match falls exactly on the English parsing boundary, and we assign a high $F_e(s, t)$ value (10 at present) to emphasize it. (1,6), (2,5), (3,5) are examples. (3,4), (4,5) are examples of *inside match*, and the value 1 is assigned to these $F_e(s, t)$ functions.

(4) [She/1 [is/2 [a/3 lovely/4 girl/5]] ./6]

[Wu 1997] introduced an algorithm to compute an optimal parsing tree for a given sentence-pair using dynamic programming (DP). This algorithm is similar in spirit to the recognition algorithm of HMM [Rabiner 1989] and to the parsing algorithm of PCFG [Lari *et al*. 1990]. The difference from the usual PCFG parsing is that the DP in SITG parses a sentence-pair simultaneously rather than a sentence only. The basic idea of DP is to divide a problem into several sub-problems, and to calculate the final solution according to the solutions of the sub-problems. In bilingual parsing, dynamic programming is used to calculate the bilingual parsing tree of a sentence-pair by decomposing it into several sub-bilingual-parsing-trees of sub-string-pairs. The whole process is that of calculating the local optimization function from the sub-parsing-tree to the whole parsing tree, recording the preceding path and back tracking along the best path in the end.

Let the input English and Chinese sentences be $e_1,...e_T$ and $c_1,...c_V$. As an abbreviation we write $e_{s...t}$ for the sequence of words $e_{s+1}, e_{s+2,}..e_t$, and similarly write $c_{u...v}$. The local optimization function $\delta(s,t,u,v) = \max P[e_{s..t} / c_{u..v}]$ denotes the maximum probability of sub-parsing-tree of node q and that both the sub-string $e_{s...t}$ and $c_{u...v}$ derive from node q. Thus, the best parser has the probability $\delta(0,T,0,V)$. In [Wu 1995b], $\delta(s,t,u,v)$ was calculated as the maximum probability combination of all possible sub-tree combinations as given below:

$$\begin{aligned}
&\delta(s,t,u,v) = \max[\delta^{[]}(s,t,u,v), \delta^{<>}(s,t,u,v)], \\
&\delta^{[]}(s,t,u,v) = \max_{\substack{s\le S\le t \\ u\le U\le v \\ (S-s)(t-S)+(U-u)(v-U)\neq 0}} \delta(s,S,u,U)\delta(S,t,U,v), \\
&\delta^{<>}(s,t,u,v) = \max_{\substack{s\le S\le t \\ u\le U\le v \\ (S-s)(t-S)+(U-u)(v-U)\neq 0}} \delta(s,S,U,v)\delta(S,t,u,U),
\end{aligned} \tag{1}$$

where S is the split point used to break $e_{s...t}$ into two constituent sub-trees, $e_{s...S}$ and $e_{S..t}$. U is the split point used to break $c_{u...v}$ into two constituent sub-trees, $c_{u...U}$ and $e_{U...v}$.The condition $(S-s)(t-S)+(U-u)(v-U)\neq 0$ serves to specify that the sub-string in one, but not both languages may be split into an empty string. Because ITG permits production in two directions, the combination of sub-trees has two corresponding directions. We use [] and <> to denote the straight and reverted production, respectively.

We integrate the constraint function $F_e(s,t)$ into the local optimization function to insert English parsing constraints in bilingual parsing. The computation of the local optimization function is modified as follows:

$$\begin{aligned}
&\delta(s,t,u,v) = \max[\delta^{[]}(s,t,u,v), \delta^{<>}(s,t,u,v)], \\
&\delta^{[]}(s,t,u,v) = \max_{\substack{s\le S\le t \\ u\le U\le v \\ (S-s)(t-S)+(U-u)(v-U)\neq 0}} F_e(s,t)\delta(s,S,u,U)\delta(S,t,U,v), \\
&\delta^{<>}(s,t,u,v) = \max_{\substack{s\le S\le t \\ u\le U\le v \\ (S-s)(t-S)+(U-u)(v-U)\neq 0}} F_e(s,t)\delta(s,S,U,v)\delta(S,t,u,U).
\end{aligned} \tag{2}$$

The other symbols in the algorithm are defined as follows: $b(e_t / c_v)$ is the probability of translating English word e_t into Chinese word c_v obtained from word alignment [Lü *et al*. 2001]. We assign a minimal probability (0.0001 at present) to empty word alignment $b(e_t / e)$ and $b(e / c_v)$. $\theta(s,t,u,v)$, $\sigma(s,t,u,v)$ and $\gamma(s,t,u,v)$ are variables used to record the production direction, the split point in English and the split point in Chinese, respectively, when $\delta(s,t,u,v)$ is achieved. These variables are used to reconstruct the bilingual parsing tree in the final step. Suppose node $q = (s,t,u,v)$; then, $\lambda(s,t,u,v) = \lambda(q)$ is the nonterminal label of q. LEFT(q) is the left sub-tree of q, and RIGHT(q) is the right sub-tree of q.

The algorithm is as follows:

1. Initialization

$$\begin{aligned}
&\delta(t-1,t,v-1,v) = b(e_t / c_v), && 1\le t\le T,\ 1\le v\le \mathrm{V} \\
&\delta(t-1,t,v,v) = b(e_t / e), && 1\le t\le T,\ 1\le v\le \mathrm{V} \\
&\delta(t,t,v-1,v) = b(e / c_v), && 1\le t\le T,\ 1\le v\le \mathrm{V}
\end{aligned}$$

2. Recursion

For all s,t,u,v ($0 \le s < t \le T$, $1 \le u < v \le \mathrm{V}$, $t - s + v - u > 2$),

$$\delta(s,t,u,v) = \max[\delta^{[]}(s,t,u,v), \delta^{<>}(s,t,u,v)],$$
$$\theta(s,t,u,v) = \begin{cases} [\,] & \text{if } \delta^{[]}(s,t,u,v) \ge \delta^{<>}(s,t,u,v), \\ <> & \text{otherwise} \end{cases}$$

where

$$\delta^{[]}(s,t,u,v) = \max_{\substack{s \le S \le t \\ u \le U \le v \\ (S-s)(t-S)+(U-u)(v-U) \ne 0}} F_e(s,t)\delta(s,S,u,U)\delta(S,t,U,v),$$
$$\delta^{<>}(s,t,u,v) = \max_{\substack{s \le S \le t \\ u \le U \le v \\ (S-s)(t-S)+(U-u)(v-U) \ne 0}} F_e(s,t)\delta(s,S,U,v)\delta(S,t,u,U),$$

$$\sigma^{[]}(s,t,u,v) = \arg_S \max_{\substack{s \le S \le t \\ u \le U \le v}} \delta(s,S,u,U)\delta(S,t,U,v),$$
$$\sigma^{<>}(s,t,u,v) = \arg_S \max_{\substack{s \le S \le t \\ u \le U \le v}} \delta(s,S,U,v)\delta(S,t,u,U),$$

$$\gamma^{[]}(s,t,u,v) = \arg_U \max_{\substack{s \le S \le t \\ u \le U \le v}} \delta(s,S,u,U)\delta(S,t,U,v),$$
$$\gamma^{<>}(s,t,u,v) = \arg_U \max_{\substack{s \le S \le t \\ u \le U \le v}} \delta(s,S,U,v)\delta(S,t,v,U).$$

3. Reconstruction

The root of the parsing tree is (0,T,0,V), and its nonterminal label is set to $\lambda(0,T,0,V) = \lambda_e(0,T)$, where $\lambda_e(s,t)$ is the English sub-tree tag that sub-string $e_{s...t}$ are derived from this sub-tree. If $e_{s...t}$ is not a sub-tree in the English parsing tree, then $\lambda_e(s,t)$ is given a tag "X". The remaining node $q = (s,t,u,v)$ in the optimal parsing tree is calculated recursively as follows:

$$\text{if } \theta(s,t,u,v) = [\,], \begin{cases} LEFT(q) = (s, \sigma^{[]}(s,t,u,v), u, \gamma^{[]}(s,t,u,v)) \\ RIGHT(q) = (\sigma^{[]}(s,t,u,v), t, \gamma^{[]}(s,t,u,v), v) \\ \lambda(\mathrm{LEFT(q)}) = \lambda(s, \sigma^{[]}(s,t,u,v), u, \gamma^{[]}(s,t,u,v)) = \lambda_e(s, \sigma^{[]}(s,t,u,v)) \\ \lambda(\mathrm{RIGHT(q)}) = \lambda(\sigma^{[]}(s,t,u,v), t, \gamma^{[]}(s,t,u,v), v) = \lambda_e(\sigma^{[]}(s,t,u,v), t) \end{cases},$$

$$\text{if } \theta(s,t,u,v) = <>, \begin{cases} LEFT(q) = (s, \sigma^{<>}(s,t,u,v), \gamma^{<>}(s,t,u,v), u) \\ RIGHT(q) = (\sigma^{<>}(s,t,u,v), t, u, \gamma^{<>}(s,t,u,v)) \\ \lambda(\mathrm{LEFT(q)}) = \lambda(s, \sigma^{<>}(s,t,u,v), \gamma^{<>}(s,t,u,v), u) = \lambda_e(s, \sigma^{<>}(s,t,u,v)) \\ \lambda(\mathrm{RIGHT(q)}) = \lambda(\sigma^{<>}(s,t,u,v), t, u, \gamma^{<>}(s,t,u,v)) = \lambda_e(\sigma^{<>}(s,t,u,v), t) \end{cases}.$$

After the bilingual parsing tree is created, the post-process consisting of rotation and flattening operations is used to restore the fanout flexibility [Wu 1997].

Using this improved SITG (ISITG), we can obtain the bilingual parsing result shown in Figure 3(b) for the given sentence-pair (3); when SBTG is used, the parsing result is that shown in Figure 2. Comparing the two results, we can see that by integrating English parsing constraints into ITG，the bilingual parsing becomes more grammatical. In the next section, we will give a quantitative experimental comparison of SBTG with ISITG.

It should be pointed out that the proposed algorithm can also be used with one-language-partial parsing, as well as with both-language parsing.

2.3 Experiments on bilingual structure alignment

To find out how important it is to include at least one language parsing, four experiments were carried out using (1) no parser (E+C); (2) only an English parser (E-parsing+C); (3) an English parser and a Chinese base phrase parser (E-parsing+C-base); (4) an English parser and a Chinese parser (E-parsing+ C-parsing). Experiment (1) followed the model of SBTG, and the other three experiments used ISITG.

The test set consisted of 2,000 English-Chinese bilingual sentence-pairs. 1,000 of the sentence pairs were collected from English textbooks for junior and senior middle school or college. The others came from the machine translation evaluation corpus of the Institute of Computational Linguistics at Peking University [Duan *et al*. 1996]. The lengths of the English sentences varied from 4 to 25 words. The test sentence pairs were first aligned at the word level based on statistics and a lexicon [Lü *et al*. 2001]. The English sentences were parsed using an incremental parser [Meng *et al*. 2001]. Both the word alignment and the English parsing were post revised manually. The Chinese parser used here is being developed by our research group. The whole parsing results are not yet robust with a precision of less than 80%. But its first stage—base phrase parsing— is quite good with a precision rate of 91.1%[Zhao *et al*. 2000]. The Chinese parsing results were not manually revised.

We evaluated the structure alignment results using a syntactic criterion. This means the matching must be grammatical. For example, for the sentence pair shown below:

(5) English: The student will get a pen .
Chinese: 这学生将得到一支钢笔。

the matchings "The student <--> 这学生", "will get<-->将得到", and "a pen <-->一支钢笔" are grammatical, while "student will<-->学生将" and "get a<-->得到一支" are ungrammatical.

All the phrases in the test set with grammatical structure matching were manually edited. These phrases were regarded as the standard structure correspondences in the evaluation. We obtained 7,812 standard structure pairs in total. The accuracy rate is defined as

$$Accuracy\ rate = \frac{standard\ structure\ numbers\ obtained\ in\ test}{total\ numbers\ of\ standard\ structures}. \quad (3)$$

Table 1. *Comparison of accuracy in bilingual structure alignment.*

Experiment type	E+C	E-parsing +C	E-parsing+C-base	E-parsing+C-pasing
Accuracy rate(%)	64.62	85.05	90.55	88.25

Table 1 shows the results of the four experiments. From the comparison of accuracy, we can see that when no parsing was conducted, the quality of alignment could not be guaranteed. The result is hardly usable for syntactic translation template acquisition. An English parsing could improve the result greatly. When a Chinese base parsing was also used, the result was even better. However, if both English and Chinese parsing were used, the result worsened slightly. This is not surprising. One reason is that Chinese parsing is still not robust. Another reason is that the two languages are parsed separately in different grammars, which may be incompatible in some respects. In the general "parse-parse-match" approach, this problem cannot be avoided.

Following is an example to illustrate the changes of the bilingual structure alignments obtained from the four experiments (Here we use the bracketing format and do not show the parsing tree in figures to save space. Readers can draw bilingual parsing trees easily according to the bracketing results.)

(6) English: This new method was brought into existence in the fifties.
Chinese: 这一新方法出现于五十年代。
English parsing: [[This new method]$_{BNP}$ [[was brought into existence]$_{VBD}$ [in [the fifties]$_{BNP}$]$_{PP}$]$_{VP}$.]$_{S}$
Chinese base phrase parsing : [这 一 [新 方法]$_{BNP}$]$_{BNP}$ 出现 于 [五十 年代]$_{BNP}$ 。
Chinese pasrsing: [[[这 一 [新 方法]$_{BNP}$]$_{BNP}$ 出现]$_{SS}$ [于 [五十 年代]$_{BNP}$]$_{PP}$ 。]$_{S}$
Result 1 (E+C): [[[[[[[[[This/这 *e*/一] new/新] method/方法] was/*e*] brought into existence/出现] in/于] the/*e*] fifties/五十] *e*/年代] ./。]
Result 2 (E-pasing+C): [[This/这 *e*/一 [new/新 method/方法]$_{BNP}$]$_{BNP}$ [[was/*e* brought

into existence/出现]$_{VBD}$ [in/于 [the/e fifties/五十]$_{BNP}$]$_{PP}$ *e*/年代]$_{VP}$ ./。]$_{S}$

Result 3 (E-parsing+C-base): [[This/这 *e*/一 [new/新 method/方法]$_{BNP}$]$_{BNP}$ [[was/e brought into existence/出现]$_{VBD}$ [in/于 [the/*e* fifties/五十 *e*/年代]$_{BNP}$]$_{PP}$]$_{VP}$ ./。]$_{S}$

Result 4 (E-parsing+C-parsing): [[[This/这 *e*/一 [new/新 method/方法]$_{BNP}$]$_{BNP}$ [was/e brought into existence/出现]$_{VBD}$]$_{SS}$ [in/于 [the/*e* fifties/五十 *e*/年代]$_{BNP}$]$_{PP}$ ./。]$_{S}$

In experiment 1, since no grammar was used, result 1 is ungrammatical. English parsing was a big help in determining the syntactic boundary of structure alignments in experiment 2. Result 2 is much better than result 1. When the Chinese base phrase parsing was also added, it helped eliminate some Chinese boundary errors(such as "[五十 年代]$_{BNP}$" in result 3). But for experiment 4, the result contradicts the English parsing result because the given Chinese parsing result is incompatible with the English parsing result.

The errors in structure alignment were mainly due to empty word alignment, where a word in one language has no counterpart string in another language. Idiomatic expressions and paraphrases usually introduce many empty word alignment errors. For example, the following two sentence-pairs, (7) and (8), can not be parsed correctly because no word is aligned in the paraphrases "has an eye ←→有鉴赏力" and "in hunger and cold ←→ 在饥寒交迫中". We can not recover these structure alignments using our algorithm for the time being.

(7) English: She has an eye for color.
Chinese: 她对颜色很有鉴赏力。

(8) English: Before liberation, peasants were struggling in hunger and cold.
Chinese: 解放前，农民在饥寒交迫中挣扎着。

Another limitation of the formalism is that it can not deal with separate two-part matches, such as the "when" match with "当.…时" in the follow example:

(9) English: Water freezes when the temperature falls below 0℃.
Chinese: 当温度下降至摄氏零度以下时，水会结冰。

It is necessary to build special productions to handle these match patterns.

Table 2. *Some examples of bilingual structure alignment.*

[<Mr./先生 Wu/吴>$_{BNP}$ <[play/拉 accordion/手风琴]$_{VP}$ [very/很 well/会]$_{ADVP}$ >$_{VP}$ ./。]$_{S}$
S[He/他 [will/将 <come/来 [in/在 [the/*e* afternoon/下午]$_{BNP}$]$_{PP}$ >$_{VP}$]./。]
[<Will/愿意 you/你>$_{X}$ [tell/告诉 me/我 [your/你的 age/年龄]$_{BNP}$ *e*/吗]$_{VP}$?/?]$_{SQ}$
[[His/他的 punishment/判刑]$_{BNP}$ <[[was/*e* commuted/减轻]$_{VBD}$ [to/为 life imprisonment/无期徒刑]$_{PP}$]$_{X}$ [by/由 [the/*e* judge/法官]$_{BNP}$]$_{PP}$ >$_{VP}$ ./。]$_{S}$
[<[We/我们 *e*/还是 had/*e* *e*/度过 *e*/了 <quite/相当 an/一个>$_{X}$ enjoyable/愉快的 holiday/假日]$_{S}$ ./， [in spite of/尽管 <the/如此 weather/气候>$_{BNP}$]$_{PP}$ >$_{S}$ ./。]$_{S}$

Some bilingual alignment results based on E-parsing+C-base are given in table 2. The syntactic structure alignments obtained with this method were later used to extract translation templates as described in the next section.

3. Translation template acquisition

When a sentence-pair is aligned using the proposed bilingual structure alignment method, the corresponding words and syntactic structures are determined. These correspondences can be used directly in translation template acquisition.

A translation template is a bilingual translation pair in which the corresponding units (words or phrases) may be replaced by variables. Two types of templates are extracted: structure translation templates and word selection templates. We take phrase or POS tag categories of noun(NN, NNS in our POS tag), verb(VB,VBP,VBZ,VBD,VBN), pronoun(PRP, PRP$)，adjective(JJ) and adverb(RB) as variables. (Our phrase symbols and POS tags are the same as those of the Penn Treebank [Marcus *et al*. 1993].)

Structure translation templates are created from phrase nodes. Each phrase node corresponds to a template. A structure translation template consists of two parts: the left side contains the component conditions of the phrase in the source language, and the right side contains the structure transfer and the translation pattern in the target language. The phrase itself is used as an index.

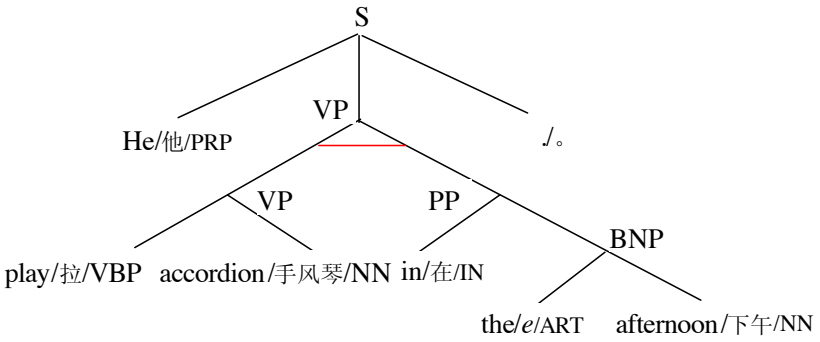


Figure 5 *A bilingual parsing tree used in translation template acquisition.*

For a bilingual structure alignment like that shown in Figure 5, five translation templates can be extracted corresponding to the five phrase nodes:

#S : 1:C=PRP+2:VP+3:W=. -> T(1)+T(2)+。;
#VP: 1:VP+2:PP ->T(2)+T(1);
#VP: 1:C=VBP+2:C=NN ->T(1)+T(2);
#PP: 1:W=in+2:BNP->在+T(2);
#BNP: 1:W=the+2:C=NN ->T(2).

The left side of the template (before ->) contains component conditions of the phrase in the source language connected with "+". "+" denotes the relation of "and", which means that the left side of the template is satisfied only when all the sub-conditions are satisfied. The numbers before ":" represent the order of the node. "W="means the word itself; "C=" means the POS category; otherwise, it is a phrase tag. The right side of the template contains the corresponding translation pattern in the target language. The function *T*(*order*) means the translation of the node "*order*". If the node is a phrase, the function returns the phrase translation by calling a structure translation template. If the node is a word, the function returns the word translation by calling a word selection template. Thus, a template "*#S: 1:C=PRP+2:VP+3:W=. -> T(1)+T (2) +* 。" means that if the phrase tag is "S" and its components satisfy the conditions that 1) the first node's category is "PRP", 2) the second node is a phrase with tag "VP" and 3) the third word is ".", then the translation should be the first node's translation plus the second node's translation, plus the punctuation mark "。". If the bilingual structure is inversely matched (with a horizontal line or "<>" notation), we write the right hand side of the template in inverse order, too. As in template "*#VP: 1:VP+2:PP->T(2)+T(1)*", the translation should be the second node's translation, followed by

the first node's translation.

It can be seen that the translation templates transfer a source structure to a target structure by changing the order of nodes on the right side. At the same time, by connecting node translation on the right side, the target translation can also be generated. Therefore, the template is a union of transfer and generalization.

The word selection template is created from the leaf node. We first get the default translation –statistically the most frequent translation in a bilingual corpus. If the current leaf node translation is not the same as the default one, we create a word selection template. For example, the word "play" has the default translation "玩" when it is a verb, while in the given example, the translation is "拉", so we get a new word selection template as follows:

#play: -1:C=PRP+0:C=VBP+1:W= accordion ->拉 .

The format of a word select template is similar to that of a structure translation template except that 1) the index entry is a word; 2) the left side of the template contains the context conditions of the word. A negative number indicates that the node is to the left of the word; 3) the right side of the template contains the translation of the word. We resolve ambiguities by adding more context words as constraints on the left side. This strategy is also used in the structure translation template.

Using the previous structure alignment corpus for the test set, we obtained a total of 7,266 templates, including 4,805 structure translation templates and 2,461 word selection templates. At present, we assume that specific templates (having the "W=" condition on the left side) have higher priority than the common templates. The frequency information of templates is also used to solve ambiguities. These acquired templates are stored in a template base. Structure translation templates and word selection templates are indexed individually by means of phrases and words. The system deals with structure translation templates and word selection templates in the same way during translation.

Translation is a recursive template matching procedure as shown in Figure 6. The input is an English parsing tree. The translation starts from the root node and works recursively top-down and from left to right. The output in the target language is generated bottom-up. It is a post-order-traverse process. When the current node is processing, all its child nodes have been processed and their translations have been determined. If no translation templates can be matched, the system uses the bilingual dictionary as the default word translation, and the structure is translated from left to right. The translation result is generated in the root node's translation field after the recursive procedure is performed.

Because the transfer and generation are combined in structure of a translation template,

the translation architecture is simpler than those of most existing translation systems, which include two separate processes for transfer and generation. The obtained translation templates are similar in format with manually edited rules, and the templates are easy to understand, so they can be modified easily and integrated into an existing machine translation system.

```
procedure Translation(ParsingTree * pnode)   // pnode is the current translation node
{
     if( IsLeafNode(pnode) )                 // decide if pnode is a leaf node
     {                                       // process for leaf node
          if ( MatchWordSelRule(pnode, rule)) //find word selection template, success return true
               pnode->translation=GetTrans(pnode, rule); //get translation according to the rule
          else
               pnode->translation=GetDefaultTrans(pnode); //get default translation
          return;
     }
     for(all pcnode, pcnode is pnode's child node )  // translate all child node
          Translation(pcnode);
     If(MatchStructureTransRule(pnode,rule))         //Find structure translation template
          pnode->translation=GetTrans(pnode, rule); //Get translation according to the rule
     else
          pnode->translation=GetDefaultTrans(pnode); //Get default translation
}
```

Figure 6 *Translation procedure.*

4. Experiments on translation using the acquired templates

In this section, we will describe translation experiments conducted based on the acquired templates to evaluate the quality of these templates.

4.1 System architecture

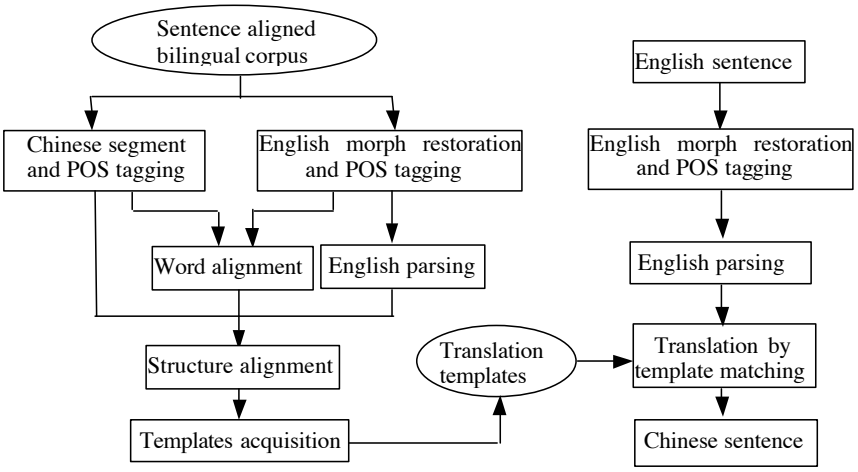


Figure 7 *System Architecture.*

An overview of the machine translation system with auto acquired translation templates is shown in Figure 7. The left part contains the learning process for translation template acquisition. The right part contains a machine translation process that uses the acquired templates. In the learning process, the bilingual sentence pairs are first aligned at the word level based on statistics and a lexicon [Lü *et al*. 2001]. Then, word alignment is extended to structure alignment as mentioned in section 2. Based on the structure alignment, translation templates are acquired and stored in a template base. In the translation process, an English sentence is parsed first; then, the template matching procedure as shown in Figure 6 is used to translate the English sentence into Chinese.

4.2 Translation experiments and evaluation

Translation experiments were conducted on the 2,000 English sentences in the test set. Some translation results and the templates used are presented in the following examples. The first line in each example is the original English sentence to be translated. The second line is the sentence' s parsing result. The last line is the Chinese translation result, and the other lines are the templates used in the translation procedure.

1) He abandoned the plan of going abroad.
[He\PRP [abandoned\VBD [the\ART plan\NN]$_{BNP}$ [of\IN [going abroad\VBG]$_{BNP}$]$_{PP}$]$_{VP}$.\FSP]$_{S}$
#S: 1:C=PRP+2:VP+3:W=. ->T(1)+T(2)+。;
#VP: 1:C=VBD+2:BNP+3:PP ->T(1)+了+T(3)+T(2);
#BNP: 1:W=the+2:C=NN ->T(2);
#PP: 1: W=of+2:BNP->T(2)+的;
#BNP: 1:W=going abroad ->出国;
#abandon: -1:C=PRP+1:W=the+2:W=plan ->放弃;
他放弃了出国的计划。

2) We passed our time pleasantly.
[We\PRP [passed\VBD [our\PRP$ time\NN]$_{BNP}$ pleasantly\RB]$_{VP}$.\FSP]$_{S}$
#S: 1:C=PRP+2:VP+3:W=. ->T(1)+T(2)+。;
#VP: 1:C=VBD+2:BNP+3:C=RB ->T(3)+T(1)+了+T(2);
#BNP: 1:C=PRP$+2:C=NN ->T(1)+T(2);
#pass: -1:C=PRP+0:C=VBD+1:W=our+2:W=time ->度过;
我们愉快地度过了我们的时间。

3) The policeman demanded his name and address .
[[The\ART policeman\NN]$_{BNP}$ [demanded\VBD [his\PRP$ name\NN and\CC address\NN]$_{BNP}$]$_{VP}$.]$_{S}$
#S: 1:BNP+2:VP+3:W=.->T(1)+T(2)+。;
#BNP: 1:W=the+2:C=NN->T(2);
#VP: 1:C=VBD+2:BNP->T(1)+T(2)
#BNP: 1:C=PRP$+2:C=NN+3:W=and+4:C=NN->T(1)+T(2)+和+T(3)
#demand: -1:W=警察+0:C=VBD+1:W=他的->询问
警察询问他的名字和地址。

To evaluate the quality of the acquired templates, we compared the translation results based on these acquired templates with those based on our existing manually edited translation knowledge base. This translation knowledge based system has the same parsing input as the learned template based system. The difference is that the system's translation process is directed by knowledge base that is totally edited by linguistic engineers. There are more than 35,000 knowledge rules in the system's knowledge base at present. The previous test set was also used as reference translation examples when the translation knowledge base was manually defined in this knowledge-based machine translation system. The evaluation followed the standards of The National High Technology Research and Development Program

(the 863 Program) machine translation evaluation project conducted in 1997 [Duan *et al*. 1996]. In the standards, translations are ranked in 6 grades, named A, B, C, D, E and F. They are defined as follows: A denotes an accurate and fluent translation; B denotes a translation that is approximately correct except for a few unimportant problems; C is a translation that can express the meaning of the source text, but some segments are ill-formed; D is a translation that is only partially correct, and separate word translations are given; E is a bad translation except that some word translations are correct; F denotes that no translation is obtained. In our evaluation, no F type translation appeared. We converted A, B, C, D and E into 100, 80, 60, 40 and 20 when calculating the average scores. 200 English sentences were random selected from test set for the manual test. These sentences were translated using the learned template-based system (LTBS) and the manually edited knowledge-based system (MEKBS), respectively. The same evaluator gave evaluations for both translations. Table 3 shows a comparison of the results. Table 4 gives some translation examples and the corresponding evaluation grades based on the acquired translation templates.

Table 3. *Translation test results.*

System \ Type	A	B	C	D	E	Average score
LTBS	60%	21%	12%	4%	3%	86.2
MEKBS	48%	41.5%	8.5%	1%	1%	86.9

The results show that without any manual encoding of translation knowledge, we were able to achieve performance nearly equal to that of traditional knowledge based machine translation. The system generated more perfect translations (A) than manually constructed translation rules did. This is because the templates were all learned automatically from real translation texts, so it could produce correct translations exactly when no ambiguities occurred. Although it also produced some bad translations (D, E), the translation results seem quite promising.

Table 4. *Some translation and evaluation grades.*

English	Translation	Grade
I will not be able to go to the movies tomorrow.	我明天不能去看电影。	A
The singer was accompanied at the piano by her pupil.	演唱者由她的学生用钢琴伴奏。	A
Which of them arrived first?	他们中哪个人第一个到达的?	A
He is having his breakfast.	他正在吃他的早饭。	B
The air here is very good.	这里空气是很好。	B
They started at night.	在晚上他们开始。	C
Will you tell me your age?	你愿意告诉我你的那个时代吗?	C
The student has a pen.	这学生长一支钢笔。	D
Some fish jump out of the water to catch insects.	一些鱼跳来自水抓住昆虫。	D
You don't like him, and I don't either.	你做也喜欢它，我做不也不喜欢。	E

Bad translations were produced because there were conflicts between templates. This disambiguation between templates is a difficult problem for any knowledge-based or example-based machine translation system. In our learning process, we solve this problem in two steps: firstly, we use the template with the highest frequency as the default template; then, when a candidate template conflicts with the default template, we add context words or categories as restrictions for this template. In the translation process, specific templates that contain a word restriction are given higher priority; otherwise the templates with highest frequency are chosen. This simple strategy works well when the training corpora are small. But when the training corpora are large, conflicts will occur more frequently. Finding a more robust method for disambiguation will be a goal of future research.

4.3 Discussion

We have developed a method for learning translation templates from bilingual corpora. These learned translation templates lead to good performance in real machine translation. Our study has shown that it is possible to reduce the need for manually encoding of translation templates, which is a difficult task in traditional knowledge-based machine translation. In addition, our method also has the following advantages:

- Compared with statistic-based machine translation(SBMT)，the translation templates obtained using our method are easier to understand than the abstract probability used by Brown [Brown *et al*. 1993].

- Unlike pure example-based machine translation (EBMT), our translation templates replace the same categories of parts-of-speech and phrases with variables, making it more general than the sentence or phrase translation examples given in [Nagao 1984].
- Unlike the traditional knowledge based (KBMT) systems, our translation templates are acquired from translation examples automatically. This can reduce the effort required for manual compilation of translation rules to a minimum.
- The learning method can easily be adapted to a new domain if only domain specific bilingual corpora are provided.

5. Conclusion and future work

Translation knowledge acquisition has been a bottleneck in machine translation. This paper has presented a method for automatic acquisition of translation templates from a bilingual corpus. The bilingual corpus is first aligned in syntactic structures using an alignment algorithm that is based on a bilingual language model and only one language parsing. The algorithm is particularly useful when a full bilingual grammar is not available. It also can be used to acquire a parsing grammar for a language lacking a well-studied grammar from a second language with a well-studied grammar. Based on the alignment result, both structure translation templates and word selection templates are extracted. Application of such templates in machine translation has demonstrated their superior performance in describing translation knowledge.

Although the results we have obtained are quite promising, there is still much to do in the near future. The corpus we used in our experiments is relatively small, and its contents are normative. We will increase the scale and extend the domain of the corpus to improve the quality and quantity of acquired translation templates. In addition, disambiguation of conflicting templates is a key problem. When the training corpus becomes large, this problem becomes serious. To solve it, we will try to introduce semantic restrictions and statistical information into templates in our future work.

Acknowledgement

This research was funded by the Microsoft Machine Translation Laboratory of the Harbin Institute of Technology. We would like to thank Microsoft Research Asia and the Institute of Computational Linguistics at Peking University for providing bilingual corpora for data training. We also would like to thank Dr. Gao Jianfeng of Microsoft Research Asia for his help in developing our ideas.

References

Hussein Almuallim, Yasuhito Akiba and Takefumi Yamazaki, "A Tool for the Acquisition of Japanese-English Machine Translation Rules Using Inductive Learning Techniques," *Proceedings of the Conference on Artificial Intelligence for Applications*, 1994，pp. 194-201.

P. F. Brown, J. C. Lai and R. L. Mercer, "Aligning Sentences in Parallel Corpora," *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, (ACL-1991)*, pp. 169-176.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra and R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, Vol. 19, No. 2, 1993, pp. 263-311.

K. W. Church, "Char-align: a Program for Aligning Parallel Texts at the Character Level," *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL-1993)*, pp. 1-8.

Ilyas Cicekli and Halil Altay Guvenir, "Learning Translation Templates from Bilingual Translation Examples," *Applied Intelligence*, Vol. 15, No. 1, 2001, pp. 57-76.

Huiming Duan and Shiwen Yu, "Report for machine translation evaluation," *Computer World*, 1996.3:183 (in Chinese)

Pascale Fung and Kathleen McKeown, "Finding Terminology Translations from Non-parallel Corpora," *The 5th Annual Workshop on Very Large Corpora*, Hong Kong: August 1997, pp. 192-202.

Ralph Grishman, and John Sterling, "Generalizing Automatically Generated Selectional Patterns," *Proceedings of 15th International Conference on Computational Linguistic (COLING-1994)*, pp. 742-747.

Halil Altay Guvenir and Ilyas Cilekli, "Learning Translation Templates from Examples," *Information Systems*, Vol.23, No. 6, 1998, pp. 353-363.

Jin-Xia Huang and Key-Sun Choi "Chinese-Korean Word Alignment Based on Linguistic Comparison," *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, pp. 392-399.

Hiroyuli Kaji, Yuuko Kida, and Yasutsugu Morimoto, "Learning Translation Templates from Bilingual Texts," *Proceedings of the 14th International Conference on Computational Linguistics (COLING-1992)*, pp. 672-678.

Sue J. Ker and Jason S. Chang, "A Class-based Approach to Word Alignment," *Computational Linguistics* 23(2), 1997, pp. 313-343.

K. Lari and S. J. Young, "The estimation of stochastic context-free grammars using the Inside-Outside algorithm," *Computer Speech and Language*, 4:35-56, 1990.

Yajuan Lü, Tiejun Zhao, Sheng Li and Muyun Yang, "English-Chinese Word Alignment Based on Statistic and Lexicon," *Proceedings of 6th Joint Symposium of Computational Linguistics*, TaiYuan, China, 2001, pp. 108-115. (in Chinese)

Christos Malavazos and Stelios Piperidis, "Application of analogical Modeling to Example Based Machine Translation," *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pp. 516-522.

M.P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a Large Annotated Corpus of English: the Penn Treebank," *Computational Linguistics* 19(2), 1993, pp. 313-330.

Yuji Matsumoto and Mihoko Kitamura, "A Machine Translation System Based on Translation Rules Acquired from Parallel Corpora," *Recent Advances in NLP,* Bulgnira 1995, pp. 27-44.

Meng Yao, Zhao Tiejun, Li Sheng and Fang Gaolin, "Incremental English parser using Combination of Statistic and Learning," *Proceedings of 863 Conference on Intelligent Computer*, Beijing, 2001, pp. 343-348. (in Chinese)

Adam Meyers, Roman Yangarber, Ralph Grishman, Catherine Macleod and Antonio Moreno. Sandoval, "Deriving Transfer Rules from Dominance-Preserving Alignments," *Proceedings of the Conference 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL-1998)*, pp. 843-847.

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

Lawrence R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, Vol. 77, No.2, February 1989. pp. 257-285.

Ralf D. Brown, "Automated Dictionary Extraction for "Knowledge-Free" Example-Based Translation," *Proceedings of the Seventh International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-1997)*, pp. 111-118.

Jung H. Shin, Young S. Han and Key-Sun Choi, "Bilingual Knowledge Acquisition from Korean-English Parallel Corpus Using Alignment Method, " *Proceeding of the 16th International Conference on Computational Linguistics (COLING-1996)*, pp. 230-235.

Davide Turcato, "Automatically Creating Bilingual Lexicons for Machine Translation from Bilingual Text," *Proceedings of the conference 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL-1998)*, pp. 1299-1305.

Hideo Watanabe, "A Method for Extracting Translation Patterns from Translation Examples," *Proceedings of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation*, 1993, pp. 292-302.

Hideo Watanabe, Sadao Kurohashi, and Eiji Aramaki, "Finding Structural Correspondences from Bilingual Parsed Corpus for Corpus-based Translation," *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pp. 906-912.

Dekai Wu, "Grammarless Extraction of Phrasal Translation Examples from Parallel Texts," *Proceedings of 6th International Conference On Theoretical and Methodological Issues in Machine Translation*, Leuven, Belgium, 1995, pp. 354-372.

Dekai Wu, "An Algorithm for Simultaneously Bracketing Parallel Texts by Aligning Words," *Proceedings of the 33th Annual Meeting of the Association for Computational Linguistics(ACL-1995)*, pp. 244-251.

Dekai Wu, "Stochastic Inversion Transduction Grammars, with Application to Segmentation, Bracketing, and Alignment of Parallel Corpora," *Proceedings of 14th International Joint Conference On Artificial Intelligence*, Montreal, 1995(IJCAI-1995), pp. 1328-1335.

Dekai Wu, "Trainable Coarse Bilingual Grammars for Parallel Text Bracketing," *Proceedings of 3rd Annual Workshop on Very Large Corpora*, Cambridge, 1994, pp. 69-82.

Dekai Wu, "Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora," *Computational Linguistics*, vol.23, No.3, 1997, pp. 377-403.

Tiejun Zhao, Muyun Yang, Fang Liu, Jianmin Yao and Hao Yu, "Statistics Based Hybrid Approach to Chinese Base Phrase Identification," *Proceedings of the 2nd Chinese Language Processing Workshop*, 2000, pp. 73-77.

Computational Linguistics and Chinese Language Processing
Vol. 6, No. 1, February 2001, pp. 109-125

Improving the Effectiveness of Information Retrieval with Clustering and Fusion

Jian Zhang[*], Jianfeng Gao[+], Ming Zhou[**], Jiaxing Wang[++]

Abstract

Fusion and clustering are two approaches to improving the effectiveness of information retrieval. In fusion, ranked lists are combined together by various means. The motivation is that different IR systems will complement each other, because they usually emphasize different query features when determining relevance and retrieve different sets of documents. In clustering, documents are clustered either before or after retrieval. The motivation is that similar documents tend to be relevant to the same query so that this approach is likely to retrieve more relevant documents by identifying clusters of similar documents. In this paper, we present a novel fusion technique that can be combined with clustering to achieve consistent improvements over conventional approaches. Our method involves three steps: (1) clustering similar documents, (2) re-ranking retrieval results, and (3) combining retrieval results.

1. Introduction

In terms of the overall performance on a large query set, none of the typical IR systems outperform others substantially, while for each individual query, the performance that different systems achieve varies greatly [Voorhees 1997]. This observation leads to the idea of combining results obtained by different IR systems to improve overall performance.

Fusion is a technique that combines retrieval results (or ranked lists) obtained by

[*] This work was done while the author worked for Microsoft Research Asia as a visiting student. Department of Computer Science and Technology of Tsinghua University, China E-mail:ajian@s1000e.cs.tsinghua.edu.cn
[+] Microsoft Research Asia E-mail:jfgao@microsoft.com
[**] Microsoft Research Asia E-mail:mingzhou@microsoft.com
[++] Department of Computer Science and Technology of Tsinghua University, China E-mail:wjx@s1000e.cs.tsinghua.edu.cn

different systems. However, conventional fusion techniques only consider retrieval results, while the information embedded in the document collection (e.g. the similarity between documents) is ignored. On the other hand, document clustering applies the structure of a document collection, but it usually considers each individual ranked list separately and is not able to take advantage of multiple ranked lists.

In this paper, we present a novel fusion technique that can be combined with clustering. Given multiple retrieval results obtained by different IR systems, we first perform clustering on each ranked list and obtain a set of clusters. We then identify the clusters that contain the most relevant documents. Each of these clusters is evaluated based on a metric called *reliability*. Documents in *reliable* clusters are re-ranked. That is, we set higher scores for these documents. Finally, a conventional fusion method is applied to combine multiple retrieval results, which are re-ranked. Our experiments on the TREC-5 Chinese collection show that the above approach achieves consistent improvements over conventional approaches.

The remainder of this paper is organized as follows. Section 2 gives a brief survey of related work. In Section 3, we describe our method in detail. In Section 4, a series of experiments are presented to show the effectiveness of our approach. Finally, we present our conclusions in Section 5.

2. Related Work

Fusion and clustering have been important research topics for many researchers.

Fox and Shaw [Fox 1994] reported on their work on result sets fusion. Their method for combining the evidence from multiple retrieval runs is based on document-query similarities in different sets. Five combining strategies were investigated, as summarized in Table 1. In their experiments, CombSUM and CombMNZ were better than the others.

Table 1. *Formulas proposed by Fox & Shaw.*

Name	Combined Similarity =
CombMAX	MAX(Individual Similarities)
CombMIN	MIN(Individual Similarities)
CombSUM	SUM(Individual Similarities)
CombANZ	$\frac{\text{SUM(Individual Similarities)}}{\text{Number of Nonzero Similarities}}$
CombMNZ	SUM(Individual Similarities) * Number of Nonzero Similarities

Thompson's work [Thompson 1990] includes assigning to each ranked list a variable weight based on the prior performance of the system. His idea is that a retrieval system should be considered preferable to others if its prior performance is better. Thompson's results were slightly better than Fox's.

Bartell [Bartell 1994] used numerical optimization techniques to determine optimal scalars (weights) for a linear combination of results. The idea is similar to Thompson's except that Bartell obtained the optimal scalars from training data, while Thompson constructed scalars based on their prior performance. Bartell achieved good results on a relatively small collection (less than 50MB).

To perform fusion more effectively, researchers began to investigate whether two result sets are suitable for fusion by examining some critical characteristics. Lee [Lee 1997] found that the overlap of the result sets was an important factor for fusion. Overlap ratios of relevant and non-relevant documents are calculated as follows:

$$R_{overlap} = \frac{R_{common} \times 2}{R_A + R_B},$$

$$N_{overlap} = \frac{N_{common} \times 2}{N_A + N_B},$$

where R_A and N_A are, respectively, the numbers of relevant and irrelevant documents in result set RL_A[1]. R_{common} is the number of common relevant documents in RL_A and RL_B. N_{common} is the number of common irrelevant documents in RL_A and RL_B.

[1] RL_A means ranked list returned by retrieval system A.

Lee observed that fusion works well for result sets that have a high $R_{overlap}$ and a low $N_{overlap}$. Inspired by this observation, we also incorporate R_{common} into our fusion approach.

Vogt [Vogt 1998, 1999] tested different linear combinations of several results from TREC-5. 36,600 result pairs were tested. A linear regression of several potential indicators was performed to determine the potential improvement for result sets to be fused. Thirteen factors including measures of individual inputs, such as average precision/recall, and some pairwise factors, such as overlap and unique document counts, were considered. Vogt concluded that the characteristics for effective fusion are: (1) at least one result has high precision/recall; (2) a high overlap of relevant documents and a low overlap of non-relevant documents; (3) similar distributions of relevance scores; and (4) each retrieval system ranks relevant documents differently. Conclusion (1) and (2) are also confirmed by our experiments, as will be shown in Section 4.3.

Clustering is now considered to be a useful information retrieval method for not only documents categorization but also interactive retrieval. The use of clustering in information retrieval is based on the Clustering Hypothesis [Rijsbergen, 1979]: *"closely associated documents tend to be relevant to the same requests"*. Hearst [Hearst 1996] showed that this hypothesis holds for a set of documents returned by a retrieval system. According to this hypothesis, if we do a good job of clustering the retrieved documents, we will likely separate the relevant and non-relevant documents into different groups. If we can direct the user to the correct group of documents, we can enhance the likelihood of finding interesting information for the user. Previous works [Cutting *et al*, 1992], [Leuski 1999] and [Leuski 2000] focused on clustering documents and let users select the clusters they were interested in. Their approaches are interactive. Most of the clustering methods mentioned above work on individual ranked lists and do not take advantage of multiple ranked lists.

In this paper, we combine clustering with fusion. Our approach differs from interactive approaches in three ways. First, we use two or more ranked lists, while others usually use one in clustering. Second, user interactive input is not needed in our approach. Third, we provide a ranked list of documents to the user instead of a set of clusters.

3. Fusion with Clustering

Our method is based on two hypotheses:

Clustering Hypothesis: Documents that are relevant to the same query can be clustered together since they tend to be more similar to each other than to non-relevant documents.

Fusion Hypothesis: Different ranked lists usually have a high overlap of relevant documents and a low overlap of non-relevant documents.

The *Clustering Hypothesis* suggests that we might be able to roughly separate relevant documents from non-relevant documents with a proper clustering algorithm. Relevant documents can be clustered into one or several clusters, and these clusters will contain more relevant documents than others. We call such a cluster a *reliable cluster*.

The *Fusion hypothesis* presents the idea of identifying *reliable clusters*. The *reliable clusters* from different ranked lists usually have a high overlap. Therefore, the more relevant documents a cluster contains, the more reliable the cluster is. We will describe the computation of *reliability* in detail in Section 3.3.

Fig.1 shows the basis idea behind our approach. Two clusters (a1 and b1) from different ranked lists that have the largest overlap are identified as reliable clusters.

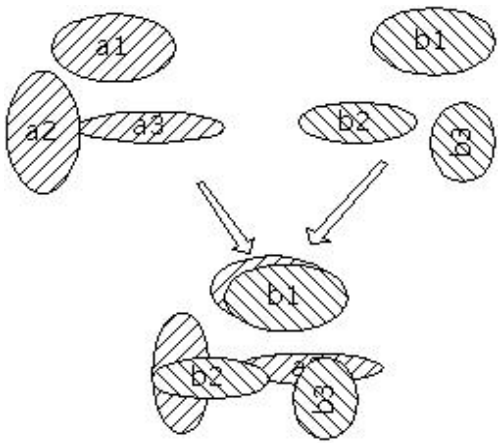


Figure 1 *Clustering results of two ranked lists.*

Our approach consists of three steps. First, we cluster each ranked list. Then, we identify the *reliable clusters* and adjust the relevance value of each document according to the *reliability* of the cluster. Finally, we use CombSUM to combine the adjusted ranked lists and present the result to user.

In the following sections, we will describe our approach in more detail. For conciseness, we will use some symbols to present our approach, which are listed in Table 2 with their explanations.

Table 2. *Notations.*

Symbol	Explanation
q	A query
d	A document
RL_A, RL_B	Ranked list returned by retrieval systems A and B, respectively
$C_{A,i}$	i th cluster in RL_A
$Sim\_CC(C_{A,i}, C_{B,j})$	Similarity between $C_{A,i}$ and $C_{B,j}$
$Sim\_qC(q, C_{A,i})$	Similarity between query q and $C_{A,i}$
$Sim\_dd(d_i, d_j)$	Similarity between two documents, d_i and d_j
$r(C_{A,i})$	*Reliability* of cluster $C_{A,i}$
$rel_A(d)$	Relevance score of document d given by retrieval system A
$rel_A^*(d)$	Adjusted relevance score of document d
$rel(d)$	Final relevance score of document d

3.1 Clustering

The goal of clustering is to separate relevant documents from non-relevant documents. To accomplish this, we need to define a measure for the similarity between documents and design a corresponding clustering algorithm.

3.1.1 Similarity between documents

In our experiments, we used the vector space model to represent documents. Each document is represented as a vector of weights $(w_{i1}, w_{i2}, ...w_{im})$, where w_{ik} is the weight of term t_k in document d_i. The weight w_{ik} is determined by the occurrence frequency of t_k in document d_i and its distribution in the entire collection. More precisely, the following formula is used to compute w_{ik}:

$$w_{ik} = \frac{[\log(f_{ik}) + 1.0] \times \log(N / n_k)}{\sqrt{\sum_j [(\log(f_{ij}) + 1.0) \times \log(N / n_j)]^2}}, \quad (1)$$

where f_{ik} is the occurrence frequency of term t_k in document d_i, N is the total number of documents in the collection and n_k is the number of documents that contain term t_k. Actually, this is one of the most frequently used *tf***idf* weighting schemes in IR.

For any two documents d_i and d_j, the cosine measure as given below is used to determine their similarity:

$$Sim\_dd(d_i, d_j) = \frac{\sum_k (w_{ik} \times w_{jk})}{\sqrt{\sum_k w_{ik}^2 \times \sum_k w_{jk}^2}}. \quad (2)$$

3.1.2 Clustering algorithm

There are many clustering algorithms for document clustering. Our goal is to cluster a small collection of documents returned by an individual retrieval system. Since the size of the collection was 1,000 in our experiments, the complexity of the clustering algorithm was not a serious problem.

Fig.2 shows our clustering algorithm. The LoopThreshold and ShiftThreshold value were set to 10 in our experiments.

Randomly set document d_i to cluster C_j;

LoopCount =0; ShiftCount = 1000;

While (LoopCount < LoopThreshold and ShiftCount > ShiftThreshold) Do

Construct the centroid of each cluster, i.e.

Centroid of $C_j = \frac{\sum_{d_i \in C_j} d_i}{|C_j|}$;

Assign d_i to its nearest cluster(the distance is determined by the similarity between d_i and the centroid of the cluster);

ShiftCount = the number of documents shifted to other cluster;

LoopCount++;

Figure 2 *Algorithm for document clustering.*

The ideal result is obtained when clustering gathers all relevant documents into one cluster and all non-relevant documents into the other cluster. However, this is unlikely to happen. In fact, relevant documents are usually distributed in several clusters. After clustering, each ranked list is composed of a set of clusters, say $C_1, C_2 \ldots C_n$.

3.1.3 Size of a cluster

The size of a cluster is the number of documents in the cluster. The clustering algorithm shown in Fig.2 cannot guarantee that the clusters will be of identical size. This causes many problems because the overlap depends on the size of each cluster.

To solve this problem, we force the clusters to have the same size using the following approach. For clusters that contain a larger number of documents than the average, we remove the documents that are far from the cluster's centroid. These removed documents are added to clusters that are smaller than average[2].

Since all the clusters are of the same size, the size of a cluster becomes a parameter in our algorithm. Thus, we need to set this parameter to an optimal value to achieve the best performance. We will report experiments conducted to determine this value in Section 4.3.

3.2 Re-ranking

After clustering each ranked list, we obtain a group of clusters, each of which contains more or less relevant documents. Through re-ranking, we expect to determine *reliable clusters* and adjust the relevance scores of the documents in each ranked list such that the relevance scores become more reasonable. To identify *reliable clusters*, we assign to each cluster a *reliability* score. According to the *Fusion Hypothesis*, we use the overlap between clusters to compute the *reliability* of a cluster. The *reliability* $r(C_{A,i})$ of cluster $C_{A,i}$ is computed as follows (see Table 2 for definitions of the symbols):

$$r(C_{A,i}) = \sum_j \left[\frac{Sim\_qC(q, C_{B,j})}{\sum_t Sim\_qC(q, C_{B,t})} Sim\_CC(C_{A,i}, C_{B,j}) \right], \tag{3}$$

where

$$Sim\_CC(C_{A,i}, C_{B,j}) = \left| C_{A,i} \cap C_{B,j} \right|, \tag{4}$$

$$Sim\_qC(q, C_{A,i}) = \frac{\sum_{d \in C_{A,i}} rel_A(d)}{\left| C_{A,i} \right|}. \tag{5}$$

[2] The size of a cluster and the number of clusters are critical issues in clustering and have been studied by many researchers. This paper focuses on how to combine fusion and clustering together and shows the potential of this combination approach. Therefore, we use a very simple method to solve the problem. Our clustering algorithm is also very simple. Our future work will be to investigate the impacts of different algorithms.

In equation (4), the similarity of two clusters is estimated based on the common documents they both contain. In equation (5), the similarity between a query and a cluster is estimated based on the average relevance score of the documents that the cluster contains. In equation (3), for each cluster $C_{A,i}$ in RL_A, its reliability $r(C_{A,i})$ is defined as the weighted sum of the similarity between cluster C_{Ai} and all the clusters in RL_B. The intuition underlying this formula is that the more similar two clusters are, the more reliable they are, as illustrated in Fig.1.

Since *reliability* represents the precision of a cluster, we use it to adjust the relevance score of the documents in each cluster. Formula (6) adjusts the relevance score of a document in a highly reliable cluster:

$$rel_A^*(d) = rel_A(d) \times [1 + r(C_{A,t})], \tag{6}$$

where $d \in C_{A,t}$.

3.3 Fusion

So far, each original ranked list has been adjusted by means of clustering and re-ranking. We next combine these improved ranked lists together using the following formula (i.e. CombSUM in [Fox 1994]):

$$rel(d) = rel_A^*(d) + rel_B^*(d). \tag{7}$$

In equation (7), the combined relevance of document d is the sum of all the adjusted relevance values that have been computed in the previous steps.

4. Experimental Results

In this section, we will present the results of our experiments. We will first describe our experimental settings in Section 4.1. In Section 4.2, we will verify the two hypotheses described in Section 3 using the results of some experiments. In Section 4.3, we will compare our approach with the other three conventional fusion methods. Finally, we will examine the impact of cluster size.

4.1 Experiment settings

We used several retrieval results from the TREC-5 Chinese information retrieval track in our fusion experiments. The document collection contains articles published in the People's Daily and news released by the Xinhua News Agency. Some statistical characteristics of the collection are summarized in Tables 3.

Table 3. *Characteristics of the TREC-5 Chinese collection.*

Number of docs	164,811
Total size (Mega Bytes)	170
Average doc length (Characters)	507
Number of queries	28
Average query length (Characters)	119
Average number of relevant docs/query	93

The 10 groups who took part in TREC-5 Chinese provided 20 retrieval results. We randomly picked seven ranked lists for our fusion experiments. The tags and average precision are listed in Table 4. It is noted that the average precision is similar except for HIN300.

Table 4. *Average precision of individual retrieval system*

Ranked list	AvP (11 pt)
BrklyCH1	0.3568
CLCHNA	0.2702
Cor5C1vt	0.3647
HIN300	0.1636
City96c1	0.3256
Gmu96ca1	0.3218
gmu96cm1	0.3579
Average :	*0.3086*

Since the ranges of similarity values of the different retrieval results were quite different, we normalized each retrieval result before combining them. The bound of each retrieval result was mapped to [0,1] using the following formula [Lee 1997]:

$$normalized\_rel = \frac{unnormalized\_rel - minimum\_rel}{maximum\_rel - minimum\_rel}.$$

4.2 Examining the hypotheses

We will first examine the two hypotheses we mentioned in Section 3.

In relation to *Clustering Hypothesis*, we clustered each ranked list into 10 clusters using our clustering algorithm. Table 5 shows some statistical information for the clustering results. The first row lists four kinds of clusters containing no, 1, 2-10 and more than 10 relevant document(s). The second row shows the corresponding percentage of each kind of cluster.

The third row shows the percentage of relevant documents in each kind of cluster.

From Table 5, we can make two observations. First, about 50% of the clusters contain 1 or no relevant document. Second, most relevant documents (more than 60%) are in a small number of clusters (about 7%). According to these observations, we can draw the conclusion that relevant documents are concentrated in a few clusters.

Thus, in our experiments, the *Clustering Hypothesis* holds in terms of the initial retrieval result when a proper algorithm is adopted.

Table 5. *Distribution of relevant docs.*

Different kinds of clusters	Containing no relevant doc	Containing 1 relevant doc	Containing 2-10 relevant docs	Containing >10 relevant docs
Percentage of each kind of cluster	38.3%	15.0%	35.0%	7.0%
Percentage of relevant docs contained in this kind of cluster	0%	3.7%	35.8%	60.5%

To test the *Fusion Hypothesis*, we computed $R_{overlap}$ and $N_{overlap}$ for each combination pair. Table 6 lists some results. The last row shows that the average $R_{overlap}$ is 0.7688, while the corresponding average $N_{overlap}$ is 0.3351. It turns out that the *Fusion Hypothesis* holds for the retrieval results we obtained.

Table 6 will also be used in Section 4.3 to confirm that $R_{overlap}$ is the most important factor determining the performance of fusion. We mark those rows whose $R_{overlap}$ scores are higher than 0.80 with the character *.

Table 6. *$R_{overlap}$ and $N_{overlap}$ values of combination pairs.*

Combination pair	$R_{overlap}$	$N_{overlap}$
BrklyCH1 & CLCHNA	* 0.8542	0.3398
BrklyCH1 & Cor5C1vt	* 0.9090	0.4393
BrklyCH1 & HIN300	0.4985	0.2575
BrklyCH1 & City96c1	* 0.8996	0.4049
BrklyCH1 & Gmu96ca1	* 0.8784	0.3259
BrklyCH1 & gmu96cm1	* 0.8871	0.3292
CLCHNA & Cor5C1vt	* 0.8728	0.4118
CLCHNA & HIN300	0.4652	0.2172
CLCHNA & City96c1	* 0.8261	0.2668
CLCHNA & Gmu96ca1	* 0.8447	0.3090
CLCHNA & gmu96cm1	* 0.8585	0.3412
Cor5C1vt & HIN300	0.4961	0.2392
Cor5C1vt & City96c1	* 0.8763	0.2943
Cor5C1vt & Gmu96ca1	* 0.9193	0.4742
Cor5C1vt & gmu96cm1	* 0.9185	0.4525
HIN300 & City96c1	0.4813	0.1555
HIN300 & Gmu96ca1	0.4636	0.1854
HIN300 & gmu96cm1	0.4701	0.2004
City96c1 & Gmu96ca1	* 0.8698	0.2854
City96c1 & gmu96cm1	* 0.8860	0.3005
Gmu96ca1 & gmu96cm1	* 0.9687	0.8064
Average	*0.7688*	*0.3351*

4.3 Comparison with conventional fusion methods

First, we studied three combination methods that were proposed by Fox, namely, CombMAX, CombSUM, and CombMNZ. Their fusion results for the same data set are listed in Table 7. The last row lists the average precision of each combination strategy. Since the average precision of the individual retrieval systems is 0.3086 (see Table 4), each of these three fusion methods has improved significantly in terms of the average precision. CombSUM appears to be the best one among them. This confirms the observation in [Fox 1994].

Then, we compared the performance of our approach with that of the other three methods, as shown in the last row in Table 7. Our new approach achieved 3% improvement over CombSUM. We also find that among all the 21 combination pairs, 17 of them are improved, compared to the results obtained using the CombSUM approach. We mark these rows with the character *.

Table 7. *Average precision of each combination pair.*

Combination pair	Comb MAX	Comb SUM	Comb MNZ	Our Approach (Cluster size=100)
BrklyCH1 & CLCHNA	0.3401	0.3627	0.3549	* 0.3755
BrklyCH1 & Cor5C1vt	0.3832	0.3976	0.3961	* 0.4107
BrklyCH1 & HIN300	0.3560	0.3243	0.2618	0.3107
BrklyCH1 & city96c1	0.3650	0.3833	0.3856	* 0.3912
BrklyCH1 & gmu96ca1	0.3753	0.4028	0.3999	* 0.4022
BrklyCH1 & gmu96cm1	0.3979	0.4234	0.4201	* 0.4243
CLCHNA & Cor5C1vt	0.3434	0.3560	0.3492	* 0.3707
CLCHNA & HIN300	0.2746	0.2478	0.2154	0.2579
CLCHNA & city96c1	0.3007	0.3459	0.3573	* 0.3931
CLCHNA & gmu96ca1	0.3269	0.3667	0.3634	* 0.3690
CLCHNA & gmu96cm1	0.3555	0.3864	0.3783	* 0.3883
Cor5C1vt & HIN300	0.3778	0.3081	0.2520	0.3139
Cor5C1vt & city96c1	0.3709	0.4091	0.4104	* 0.4285
Cor5C1vt & gmu96ca1	0.3568	0.3684	0.3676	* 0.3724
Cor5C1vt & gmu96cm1	0.3831	0.3926	0.3911	* 0.3975
HIN300 & city96c1	0.2616	0.2565	0.2444	0.3036
HIN300 & gmu96ca1	0.3466	0.2942	0.2464	0.2954
HIN300 & gmu96cm1	0.3764	0.3205	0.2613	0.3150
city96c1 & gmu96ca1	0.3310	0.3764	0.3854	* 0.3939
city96c1 & gmu96cm1	0.3595	0.3970	0.4047	* 0.4090
gmu96ca1 & gmu96cm1	0.3451	0.3514	0.3511	* 0.3505
Average:	*0.3489*	*0.3557*	*0.3426*	*0.3654*

Comparing the results shown in Table 7 with those listed in Table 6, we find that the pairs with a $R_{overlap}$ of over 0.80 correspond to better combination performance. We call this kind of pair a *combinable pair*. For example, BrklyCH1 & CLCHNA is a *combinable pair*. Although the average combination performance is 0.3654 (using our approach), almost all the *combinable pairs* exceed the average performance[3]. This again confirms the conclusion in both [Lee 1997] and [Vogt 1998] that the performance of fusion heavily depends on $R_{overlap}$. It also reveals the limitation of our approach and of other linear fusion techniques in that a high overlap of relevant documents is a pre-requisite for performance enhancement. For those pairs that don't satisfy this pre-requisite, normal fusion may even decrease retrieval performance.

We also compared our approach with the optimal linear combination. Since ranked lists

[3] "gmu96ca1 & gmu96cm1" is an exception because their related $N_{overlap}$ score is very high.

are combined linearly, only the ratio of the two weights affects the final performance:

$$RL_{combined} = RL_A + wRL_B.$$

CombSUM can be taken as a special case of linear combination where w is set to be 1. When the relevant documents are known, the weight w can be optimized using some numerical method. In our experiment, the weight w was optimized using golden section search [Press 1992]. This approach was adopted in [Vogt 1998]. The average precision for the optimal linear combination we obtained is 0.3714. As shown in Fig.3, our approach performs better than CombSUM and CombMAX and is very close to CombBest.

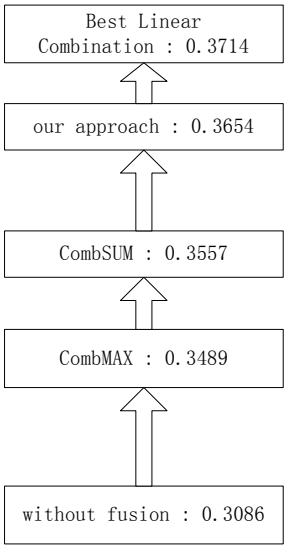


Figure 3 *Performance of different approaches.*

To summarize, we can draw three conclusions from the above experiments. First, in most cases, our new approach shows better performance than most of the conventional methods, including CombSUM and CombMNZ. Second, $R_{overlap}$ strongly affects the performance of linear fusion. Third, the performance of our approach is very close to that of the optimal linear combination approach.

4.4 Impact of cluster size

We also studied the impact of cluster size. Table 8 shows the experimental results. When the cluster size varied from 200 to 5, the average precision did not change much. The maximum value was 0.3675 when the cluster size was 25 and the minimum value was 0.3621

when the cluster size was 200. This shows that the cluster size setting has very little impact in our approach.

Table 8. *Impact of cluster size.*

Size of Cluster	200	100	50	25	10	5
11pt AvP	0.3621	0.3654	0.3661	0.3675	0.3668	0.3661

Another interesting question is what will happen when the cluster size is set to 1000 or 1.

When the cluster size is set to 1000, each ranked list becomes a single cluster. Then, the reliability of C_A and C_B can be computed as follows:

$$r(C_A) = r(C_B) = Sim\_CC(C_A, C_B) = |C_A \cap C_B|.$$

Since $r(C_A)$ and $r(C_B)$ are equal, the re-ranking and fusion step becomes a normal CombSUM approach, and the average precision is equal to that of the CombSUM approach.

When the cluster size is set to 1, each document forms a cluster by itself. Those documents appearing in both ranked lists will be improved. For those documents that only appear in one ranked list, their relevance will remain unchanged. On the other hand, the relevance score of those documents that appear in both ranked lists will be improved with a factor of $1 + \frac{Sim\_dd(q,d)}{\sum Sim\_dd(q,d_j)}$. The final result will be close to that of the CombSUM approach because this factor is close to 1.

The impact of the cluster size setting is illustrated in Fig.4. From this figure, we find that fusion combined with clustering is consistently better than the approaches that do not include clustering (where cluster size = 1000). We find that a setting size to 25 gives the best combination when the ranked list has a size of 1,000.

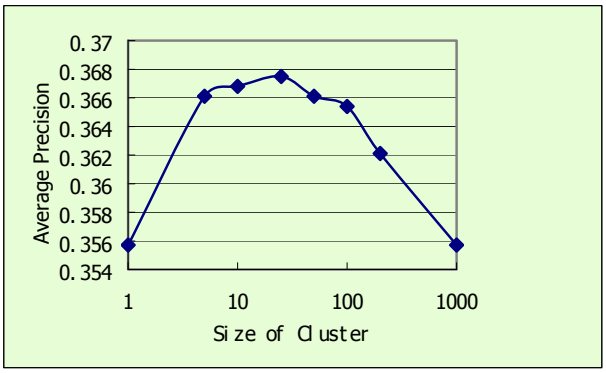


Figure 4 *Impact of cluster size.*

5. Conclusion

Combining multiple retrieval results is certainly a practical technique for improving the overall performance of information retrieval systems. In this paper, we have proposed a novel fusion method that can be combined with document clustering to improve retrieval performance. Our approach consists of three steps. First, we apply clustering to the initial ranked document lists to obtain a list of document clusters. Then, we identify reliable clusters and adjust each ranked list separately using our re-ranking approach. Finally, conventional fusion is carried out to produce an adjusted ranked list.

Since our approach is based on two hypotheses, we first verified them by means of experiments. We also compared our approach with other conventional approaches. The results show that each of them achieves some improvement, and that our approach compares favorably with them. We also investigated the impact of cluster size. We found that our approach is rather stable under variation in the size of clusters.

Although our method showed good performance in our experiments, we believe it still can be improved further. A better clustering algorithm for identifying more reliable clusters and more elaborate formula for re-ranking ranked lists should lead to further improvement. These will be topics for our future work.

References

Bartell,B.T., Cottrell,G.W., and Belew,R.K., "Automatic Combination of Multiple Ranked Retrieval Systems," *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1994, pp. 173-181.

D.R.Cutting, D.R.Karger, J.O.Pedersen, and J.W.Tukey, "Scatter/gather: A Cluster-based Approach to Browsing Large Document Collections," *Proceedings of the 15th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1992, pp. 126-135.

Fox,E. and Shaw,J., "Combination of Multiple Searches," The Second Text Retrieval Conference (TREC2), NIST Special Publication 500-215, 1994, pp. 243-252.

Hearst,M.A., and Pedersen,J.O., "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results," *Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1996, pp. 76-82.

J.H.Lee. "Analyses of Multiple Evidence Combination.," Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1997, pp. 267-276.

A.Leuski and J.Allan, "The Best of Both Worlds: Combining Ranked List and Clustering," *CIIR Technical Report IR-172*, 1999, http://cobar.cs.umass.edu/pubfiles/ir-172.ps.

A.Leuski and J.Allan, "Improving Interactive Retrieval by Combining Ranked List and Clustering," Proceedings of RIAO(Recherche d'Informations Assistee par Ordinateur = Computer-Assisted Information Retrieval) 2000 Conference, 2000, pp. 665-681.

C.J.van Rijsbergen, *Information Retrieval*, Butterworths, London, second edition, 1979.

Thompson,P., "A Combination of Expert Opinion Approach to Probabilistic Information Retrieval, part I: The Conceptual Model," *Information Processing and Management*, 26(3) 1990, pp. 371-382.

Vogt,C., Cottrell,G., Belew,R. and Bartell,B., "Using Relevance to Train a Linear Mixture of Experts," *Proceedings of the 5th Text Retrieval Conference (TREC5), NIST Special Publication 500-238*, 1997, pp. 503-516.

Vogt,C. and G.Cottrell., "Predicting the Performance of Linearly Combined IR Systems," Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1998, pp. 190-196.

Vogt,C. and Cottrell,G., "Fusion Via a Linear Combination of Scores," *Information Retrieval*, 1(2-3), 1999, pp. 151-173.

E.Voorhees, D.Harman, "Overview of the Sixth Text Retrieval Conference (TREC-6)," *NIST Special Publication 500-240*, 1997. pp. 1-24.

Press,W.H., Teukolsky,S.A., Vetterling,W.T., and Flannery,B.P., *Numerical Recipes in C - The Art of Scientific Computing*, Cambridge University Press, 1992.