University of Alberta

A TIGHTNESS CONTINUUM MEASURE OF CHINESE SEMANTIC UNITS, AND ITS APPLICATION TO INFORMATION RETRIEVAL

by

Ying Xu

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Spring 2010
Edmonton, Alberta

Examining Committee

Randy Goebel, Computing Science

Christoph Ringlstetter, Center for Language and Information Processing, University of Munich (LMU)

Greg Kondrak, Computing Science

Dangzhi Zhao, School of Library and Information Science

To my parents.

献给我亲爱的爸妈.

Abstract

Chinese is very different from alphabetical languages such as English, as there are no delimiters between Chinese words. So Chinese segmentation is an important step for most Chinese natural language processing (NLP) tasks such as machine translation (MT) and information retrieval (IR). Previous work has shown a non-monotonic relation between improvements in Chinese segmentation performance and performance on NLP tasks. Our research also suggests that different tasks need different criteria for Chinese segmentation.

We propose a tightness continuum for Chinese semantic units which provides a more principled approach to the coupling of segmentation methods and NLP application tasks. The construction of the continuum is based on calculating the frequency distribution of units' segmentation patterns. For a Chinese character sequence of length n, 2^{n-1} potential segmentation candidates exist. Based on this continuum, sequences can be dynamically segmented, and then that information can be exploited in a number of information retrieval tasks.

In order to show that our tightness continuum is useful for NLP tasks, we propose two methods to exploit the tightness continuum within IR systems. The first method refines the result of a general Chinese word segmenter: it combines units which are tightly connected according to statistical information but segmented by the former segmenter, and segments units which are not tight but previously treated as one unit. The second method embeds the tightness value into IR score functions according to our hypothesis that terms in tight queries are more likely to be consecutive in relevant documents than terms in loose queries. After analyzing the currently available Chinese test collections, we found that they are not suitable for evaluating the effects of Chinese segmentation, especially the segmentation of Chinese compounds, on IR. So we created a focused test collection. Experimental results show that our tightness measure is reasonable and does improve the performance of IR systems. As another consequence our experiments demonstrate a strong need for additional corpora for the investigation of Chinese IR.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisors Prof. Randy Goebel and Dr. Christoph Ringlstetter. Randy introduced me with this challenge yet exciting topic, most importantly, lead me to find my way of how to do research. I just entered into this Neverland and I wish I could be a tinker here. Christoph gave me a lot of precious advice, and made me realize the importance of result data and what to look for into data after experiments.

I would like to thank Prof. Greg Kondrak, PhD student Shane Bergsma, PhD student Jiyang Chen, Dr. Christopher Pinchak, Shen Jiang, Yifeng Liu, and Qing Dou for their assistance at various stages of this research.

Finally, I would like to thank my parents and my friends, who supported me during the difficult time of study abroad.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Chinese is very different from alphabetical languages such as English, as there is no delimiter between Chinese words. There are those who believe that Chinese "does not have words" but instead has "characters." But we agree with (Packard, 2000), that "speakers of Chinese compose and understand sentences just as speakers of any languages do, by manipulating sentence constituents using rules of syntax; the smallest representatives of those constituents have the size, feel, shape and properties of words." So Chinese word segmentation is preliminary to most Chinese text processing tasks. For example, the sentence "研究中文分词在信息检索中的作用" (Analyzing the effects of Chinese text segmentation for information retrieval) needs to be segmented into "研究(Analyzing) | 中文(Chinese) | 分词(segmentation) | 在(upon) | 信息(information) | 检索(retrieval) | 中(in) | 的('s) | 作用(effects)." In the former sentence and the following, "|" is used as a segmentation mark. If we take away the delimiters in English, the sequence becomes "Analyzingtheeffects," and we have to segment it into "Analyzing | the | effects." Intense work has been done to analyze the effects of word segmentation on different tasks such as *information retrieval* (IR) and *machine translation* (MT). It has been recognized that different NLP applications have different needs for segmentation (Chang *et al.*, 2008). For example, it is intuitive that IR tasks prefer a more fine-grained segmentation than MT tasks. In this dissertation, we will concentrate on the effects of Chinese word segmentation for IR.

Broadly speaking, the information retrieval task focuses on using text queries to retrieve information from documents. One classical problem in IR is the *ad-hoc retrieval problem*. In ad-hoc retrieval, the user enters a query describing the desired information. The system then returns a list of documents related to the query (Manning and Schutze, 1999). The retrieved documents are ranked according to the correlation between query terms and document terms. To speed up, terms in documents are organized in an *inverted index*, where the keys are terms, the values are lists of documents containing the terms. Our application of Chinese compound analysis will focus on ad-hoc retrieval.

In general two design methods exist to segment documents and queries for Chinese IR: a character based method, which takes characters directly as index and query terms; and a word-based method, which requires word segmentation as a pre-step. Most published results showed unigram

indexing to be inferior to word based indexing, while bigram indexing is superior, with hybird methods yielding the best results. In the terminology of *n-gram*, *gram* in Chinese means Characters.

One popular form of an information retrieval system is a Search Engine. There are many Chinese Search Engines, such as Baidu (www.baidu.com), Google (www.google.cn), and Sogou (www.sogou.com). After a series of experiments, we found none of them segments web pages when building the index, but some of them segment queries. Sometimes the result turns out to be nonsense as the word segmentation is not taken into account, since the meaning of a non-compositional word is not related to the meaning of its components. For example, for querying "非国大" (African National Congress) in Google, one result is "此'茅台'非'国酒茅台"'(this 'Maotai' is not that 'national wine Maotai'). In this example, the meaning of the word "非国大" (African National Congress) is not related to the meaning of its components, "非" (not), "国" (country), and "大" (big). The search engine segmented the query into three characters. The false result was retrieved since it contains two characters of the three.

Our hypothesis is that appropriate word segmentation for IR should improve Chinese IR performance. While most Chinese word segmentation methods employed in IR systems are based on a static dictionary or a machine learning model which was trained upon a manually segmented corpus, Chinese IR systems need segmentation that is based on the semantics of compounds. So in this research, we investigate what Chinese segmentation is suitable for Chinese IR.

There are two major components in our research. First, we proposed the hypothesis that Chinese semantic units do not fall cleanly into the binary classes of compositional or non-compositional, but into a continuum of tightness and looseness, where tightness is considered as a degree of compositionality. Then we designed a tightness measure of Chinese units based on the frequency distribution of segmentation patterns. *Compositionality* means the extent to which the meaning of a string with multi components is related to the meaning of its components. According to this tightness continuum, we developed a segmentation method which combines units that are tightly connected according to statistical information, and segments units which are not tight. Second, we need to evaluate the effect of this tightness continuum for IR. In this step, we propose two methods to exploit the tightness measure in Chinese IR systems. In addition, we create our own test collection as the current collections are neither suitable nor large enough for a deep analysis of the effects Chinese segmentation has upon IR.

In the following sections, a more specific introduction of these two steps is presented. The first section is the introduction of a specific measure for the tightness of Chinese linguistic units. In the second section we investigate the effects of this measure upon Chinese IR. In the last section of this chapter, the outline of this thesis will be presented.

1.1 Tightness Measure

As previously mentioned, Chinese word segmentation is the preprocessing step for information retrieval. There are several ways to segment Chinese sentences:

- 1. dictionary based methods which need a lexicon;
- 2. machine learning methods which take a corpus, a dictionary, part of speech tags (POS), and so on for training.

To some extent, Chinese word segmentation is similar to multiword expression extraction in English, both of which try to extract multi-gram semantic units. The difference is that "gram," i.e. "sociological word," is the familiar "word" in English, compare to the "character" in Chinese (Packard, 2000). *Multi-gram extraction* identifies units such as "kick the bucket," "at gunpoint," or "make out" in English, and units such as "花生" (peanut), "月下老人" (match maker), or "乌鲁木齐" (Urumchi, name of a city) in Chinese. It is intuitive that character sequences that are tightly connected, or non-compositional, i.e. the meaning of the whole word can not be predicted from the meaning of its components, should be kept as one unit, while others should be segmented. But the boundary between what is non-compositional and compositional is difficult to draw, which creates ambiguity for word segmentation. For example, it is certain that "上海哪有" (Where in Shanghai) is compositional and should be segmented into "上海" (Shanghai) "哪有" (where). We are sure that "月下老人" (match maker) is non-compositional and should be kept as one word. But what about "机器学习" (machine learning)? Is it compositional or non-compositional? Should it be segmented into "机器" (machine) and "学习" (learning) or kept as one word? This ambiguity is also one of the reasons why we believe former word segmentation models trained on manually segmented data or dictionary based methods are not suitable for Chinese IR. The boundaries are set abruptly without concern for the characteristics of different corpora and the meaning of words. Supporting evidence for our belief is that, in (Peng *et al.*, 2002), they suggested that the relationship between Chinese word segmentation and IR is non monotonic; better segmentation, for which manually segmented data are employed as evaluation data, does not always yield better IR performance. Their analysis found that one reason was that manual segmentation standard kept some compounds as words, while segmenting them into components yielded better results.

Our hypothesis is that independent Chinese semantic units (also referred to as "Chinese sequences" in the following) as observed in a text do not fall cleanly into the binary classes of compositional or non-compositional, but into a continuum of tightness and looseness. Intuitively, this continuum also exists in naturally segmented languages such as English (Halpern, 2000). This tightness characteristic of units determines their linguistic nature as well as their preferred treatment in different NLP applications, e.g., for two consecutive nouns, whether to index two nouns or one nominal compound in IR, or to translate them as a unit or separately in MT. For different NLP ap-

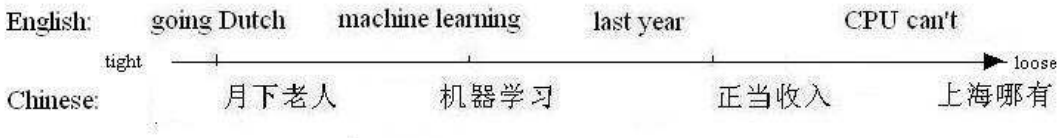

Figure 1.1: The continuum of tightness

plications, the threshold for how tight a Chinese unit needs to be so that we keep it as a word will be different, but binary classification of semantic units is not enough.

Along this tightness continuum, at one extreme are non-compositional semantic units, such as idioms, non-compositional compounds, and transliterated names; at the other end are *purely consecutive words* which means there is no dependency relation between those words, with compositional compounds and phrases in between. Figure 1.1 shows some examples of English and Chinese multigram semantic units along this tightness continuum, where the left end is the tightest and the right end is the loosest. For English, "going Dutch" is a non-compositional idiomatic expression as its meaning has nothing to do with combination of the literal meanings of "going" and "Dutch"; the same holds for "milky way," a non-compositional compound; "machine learning" is a compositional compound but a tight one as compared to "plum pie" which is significantly looser; "last year" is a common sense phrase with "last" as a modifier of "year"; "CPU can't" is a phrase in a text with an arbitrary nominal CPU preceding the very general modal "can." For Chinese, "月下老人" (match maker) is a non-compositional idiomatic expression since its meaning has nothing to do with combination of the literal meaning of "月下" (under the moon) and "老人" (old people); "乌鲁木齐" (Urumchi) is a non-compositional transliterated proper noun; "机器学习" (machine learning) is a compositional compound; "正当收入" (legitimate income) is a phrase; and "上海哪有" (Shanghai where) are two consecutive words.

Many methods have been proposed to measure non-compositionality of units. One popular example is *pointwise mutual information* (PMI), which uses the frequency of terms in the document. More details of the MI method and differences to our measure are given in Chapter 2. In our work, we exploit corpus data and propose a method to locate a Chinese semantic unit in the continuum of tightness and looseness. The input to our approach are document frequencies of segmentation patterns for semantic units, i.e. number of documents that contain a specific segmentation pattern. A pattern is a potential segmentation, which here means that a character sequence of length n has 2^{n-1} different patterns. For example, "机器学习," "机|器学习" and "机器|学习" are possible segmentation candidates for "机器学习" (machine learning). Note that every pattern contains all the characters of the unit. The intuition of using document frequency is that a document containing all the characters of a unit provides a stronger basis for the semantics of that unit than a document containing fewer characters.

We confirmed that our measure does capture the tightness of Chinese semantic units with two experiments. First, we used our tightness measure to rank 300 Chinese semantic units according to their tightness and compare the result with a manually created gold standard ranking. The evaluation showed that the automatic ranking is comparable to the manual ranking. Second, we extracted non-compositional semantic units from the Chinese Gigaword corpus, which contains more than 1 Gigabyte of Chinese text, and compared the result with a Chinese dictionary. The precision is promising, which further supports the value of our tightness measure. In addition, we used our tightness measure to segment the *Chinese Treebank*, and obtained a promising result. The *Penn Chinese Treebank* is a sgemented, part-of-speech tagged, and fully bracketed corpus that currently has 500 thousand words (over 824K Chinese characters). More details of the tightness continuum are given in Chapter 3.

1.2 Information Retrieval

The goal of IR is to retrieve information from document repositories. Chinese IR systems have the same structure as other languages' IR systems. The only difference is the segmentation pre-step. In the following, We will first provide some background of information retrieval systems, and then present the introduction to our research.

1.2.1 Background

The most important task for IR is how to retrieve documents that are related to the query. Most systems get a list of documents sorted according to the extent of the correlation between the documents and the query. In the following a simple structure of the employed IR framework is described. A segmenter model, i.e. a token extraction model, segments the query into several terms, e.g. English words or Chinese words; in order to speed up the process, an *inverted index* of terms in the corpus is built; and a score function calculates the score of every document in the corpus corresponds to the query. How to weight documents and rank them is decided by the score functions. Figure 1.2 shows the main components of this structure.

The *inverted index* is the most important part of IR systems. It is a list of terms, with each term having a list of documents containing the term and its *term frequency*.

To measure the correlation between a term and a document, i.e., to rank the documents, many different score functions for document relevance ranking have been proposed, such as tf*idf and BM25. Before giving the details of these two functions, several terms need to be defined.

Term frequency (TF): The term frequency in a given document d_j is the number of times a given term t_i appears in the document. It is a measure of the importance of the term t_i within the particular document d_j.

$$TF(t_i, d_j) = \frac{freq(t_i, d_j)}{\sum_{t_k \in d_j} freq(t_k, d_j)} \tag{1.1}$$

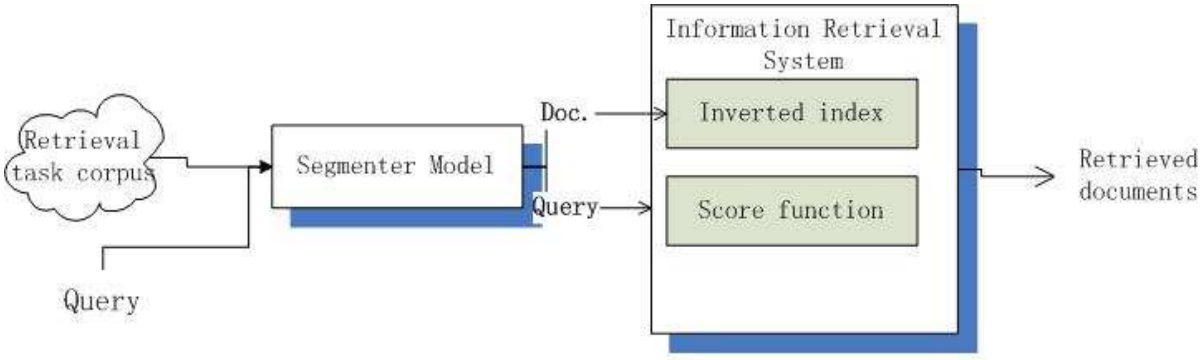

Figure 1.2: The IR system structure

where $freq(t_i, d_j)$is the frequency of the term t_i in the document d_j.

Inverse document frequency (IDF): The inverse document frequency is a measure of the general importance of the term t_i over a given corpus D. It is defined as

$$IDF(t_i, D) = log\frac{|D|}{|\{d_j|t_i \in d_j\}|} \tag{1.2}$$

where $|D|$ is the number of documents in the corpus. The IDF weight was first introduced in (Jones, 1972) as a tuning parameters for TF. It gives penalty to those too frequent words, such as function words. There is a log in the function because they assumed Zipf shape for the document frequency.

The popular score function tf*idf is the production of TF and IDF.

Another function BM25 is in the Equation 1.3 (Robertson *et al.*, 1994).

$$BM25(t_i, d_j) = \frac{freq(t_i, d_j)}{k((1-b) + b\frac{l_{d_j}}{avl_d}) + freq(t_i, d_j)} * IDF'(t_i, D) \tag{1.3}$$

where $IDF'(t_i, D)$ is a little different from the former IDF,

$$IDF'(t_i, D) = log\frac{|D| - |\{d_j|t_i \in d_j\}| + 0.5}{|\{d_j|t_i \in d_j\}| + 0.5}; \tag{1.4}$$

l_{d_j} is the document d_j length; avl_d is the average document length over the corpus D; k is a free parameter (usually 2); and $b \in [0, 1]$ (usually 0.75).

The evaluation of IR systems is based on the recall and precision of retrieved documents comparing with standard relevant document set. The *recall* is the ratio of the number of relevant documents retrieved to the number of total relevant documents (Equation 1.5). The *precision* is the ratio of the number of relevant documents retrieved to the number of documents retrieved (Equation 1.6).

Both recall and precision need to be taken into concern for evaluation. Typical comprehensive measures include *F-score* (Equation 1.7), *average precision* (Equation 1.8), or interpolated precision (e.g. mean precision for points at recall 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1).

$$Recall = \frac{\text{the number of relevant documents retrieved}}{\text{the number of total relevant documents}} \tag{1.5}$$

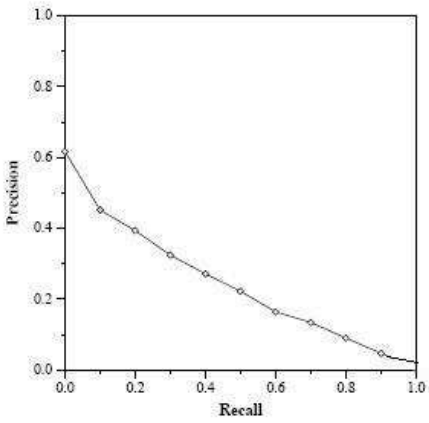

Figure 1.3: An example of the recall and precision curve

$$Precision = \frac{\text{the number of relevant documents retrieved}}{\text{the number of documents retrieved}} \tag{1.6}$$

$$F - score = 2\frac{precision \times recall}{precision + recall} \tag{1.7}$$

$$AP = \frac{1}{|R|}\sum_{i} precision@rank(i) \tag{1.8}$$

where $i \in R$ is a relevant document, $|R|$ is the total number of relevant documents, and $precision@rank(i)$ is the precision at the point where the relevant document i is retrieved. The data for evaluation is a *test collection*. There are commonly three components in a test collection, the query set, the document set, and the relevant document set, which is a subset of the former. Here is an example for IR evaluation. The evaluation data contains a query set with only one query Q, a document set {d1, d2, d3, d4, d5, d6, d7}, and the relevant document set is {d1, d2, d3, d5, d7}. An IR system S extracts documents in the following order, d1, d2, d3, d4, d5, for the query Q. The recall of the system is $4/5$ as there are four relevant documents retrieved out of 5. The precision is $4/5$, as 4 documents are relevant out of the 5 retrieved documents. So the F-score is $2 * \frac{4}{5} * \frac{4}{5} * (\frac{4}{5} + \frac{4}{5})$. The Average Precision is $\frac{1}{5} * (\frac{1}{1} + \frac{2}{2} + \frac{3}{3} + \frac{3}{4} + \frac{4}{5})$. If it retrieves only the first three documents, then the precision is 1, while the recall is $3/5$. We can tune a system to balance between the value of recall and precision. It depends on different tasks about which is more important, high recall or high precision. For example, for search engines, high precision is more important, as there are enough relevant documents and users will only check the first several results. Usually, there will be many queries, and then we calculate the mean for every metric. Figure 1.3 shows an example of the recall and precision curve. Usually, the relationship is inversely proportional, i.e. precision decreases as recall increases.

1.2.2 Chinese Information Retrieval

Chinese information retrieval (IR) systems employ the same framework as those of English. What distinguishes Chinese IR and others is the segmenter model, i.e. how to segment queries and documents. Usually, documents and queries have to be segmented in the same manner as this increases the probability of the query-document match. There are several ways to segment a Chinese sentence, based on unigrams, on overlap bigrams, and on words. For example, for a sequence "ABCD," unigram index method will get {"A", "B", "C", "D"}; overlap bigram will get {"AB", "BC", "CD"}; and word segmentation will get {"A", "B", "CD"}, if the dictionary contains {"CD"}. As aforementioned, much research has concentrated on analyzing the effects of word segmentation on Chinese information retrieval (Nie *et al.*, 2000; Foo and Li, 2004; Peng *et al.*, 2002; Shi and Nie, 2009). They compared different segmentation methods in different IR systems. But no deep explanations are provided about why one segmentation method is better than the other, or why the bigram method has such an un-intuitively good result (bigram gets the best result in most experiments presented so far). We believe one difficulty of analyzing the results more deeply is the lack of data.

We propose two methods to apply our semantic tightness measure within IR systems. The first method refines the results of a general Chinese word segmenter, e.g., ICTCLAS, a segmenter provided by Institute of Computing Technology, Chinese Academy of Science. Our method combines units which are tightly connected according to statistical information but segmented by the former segmenter, e.g., combining "乌"(black)"鲁"(lu) "木"(wood) "齐"(parallel) into "乌鲁木齐" (Urumchi). It segments units which are not tight but treated as one unit before, e.g., segmenting '科威特国" (Country Kuwait) into "科威特"(Kuwait) "国"(country). The second method embeds the tightness continuum into IR score-functions according to our hypothesis that terms in tight queries are more likely to be in a consecutive form in relevant documents than loose queries.

The most popular Chinese IR evaluation data sets, the sets of the *Text REtrieval Conference* (TREC), are produced by the *National Institute of Standards and Technology* (NIST). Every data set contains a set of queries, a corpus, and a set of relevant documents from the corpus for each query, which are determined by human annotators. Most work on Chinese IR evaluates competing systems with TREC 5, 6 Chinese tracks, which together provide only 54 queries that are organized with relevancy-judged documents. During our experiments, we found the TREC query data is ill-suited for analyzing the effects of Chinese segmentation, especially compound segmentation, on IR. For this reason, we created an additional set of queries which range from tight to loose, while retaining the TREC corpus as the document base. Since we currently lack resources to conduct pooling experiments, as have been conducted by NIST for TREC, we have employed the minimal test collection method as introduced in (Carterette *et al.*, 2006).

Our experiments showed that the first method, segmentation refinement, does significantly improve IR performance. For the second method, the improvement of the adapted score function turned out to be independent of our tightness value, but the result gives evidence that for Chinese

IR, it is better to segment documents in a more fine grained way while combining terms through retrieval progress, e.g. choosing a score function which prefers short term distances. Details of the two methods are given in Chapter 4.

1.3 Outline

This document is organized as follows. Chapter 2 gives the description of related work. Chapter 3 presents our approach to measure the tightness of units built from units' segmentation pattern distribution in a corpus. Chapter 4 describes integration of the tightness measure into an IR system. Chapter 5 summarizes our work, suggests possible future directions, and concludes this dissertation.

Chapter 2

Related Work

There are two main related fields for this research. One is Chinese word segmentation, the other is information retrieval. As it was pointed out earlier, Chinese word segmentation is closely related to English multi-word extraction. So in the following, we first present several background aspects of English multi-word extraction. We describe several models of Chinese word extraction from text, while some items of interest and the difference between those work and our methods were pointed out. Next we summarize previous work which analyzed the effect of word segmentation (not only in Chinese) on information retrieval. Finally, we introduce past methods which tried to reduce the resources spent on IR evaluation data, one of which has been employed to create our data set.

2.1 English Multi-word Extraction

Many people are working on acquisition of multi-word expressions, although the terminology varies. Most linguists call it multi-word extraction (McCarthy *et al.*, 2003), while computer scientists call it collocation extraction (Manning and Schutze, 1999). Much of these work has proposed measures for the tightness characteristics of multi-word expressions, e.g. pointwise mutual information, likelihood ratio, and chi-square (Manning and Schutze, 1999).

Linguists have been interested in multi-word expression for a long time. They especially focused on how the multi-word expression should be treated for dictionary construction. For example, (Guenthner and Blanco, 2004) discussed the treatment of different complex lexical items, such as compounds, collocations, idioms, frozen expressions, for building a dictionary. Two issues discussed were: 1) the nature of items, i.e. their place in dictionary, postulation of the categories into which they should be classified, and the type of information that should be attached to them; 2) the state of the art of structuring the set of items in such a way that their properties (morphological, syntactic, semantic) can be of direct use in analyzing other items. (McCarthy *et al.*, 2003) investigated various statistical measures of compositionality of candidate multiword verbs, specifically English phrasal verbs identified automatically using a robust parser. These measures compared the nearest neighbors of the phrasal verbs to the neighbors of the corresponding simplex verb. For example, compare the

neighbor of "climb down" with those of "climb." The intuition is that the more compositional the phrasal, the closer will be the neighbors of the phrasal and the corresponding simplex verb. They compared the ranking of 111 verb phrases by a variety of statistical measures with that ranked by human annotators. Note that their work depended on the result of automatic part of speech (POS) tagging and a synonym list, while our method takes raw corpus data as input directly. The best result they got is a correlation of 0.49 with human annotators. While they ranked their test phrase set on a 10 rank scale, we ranked them on a 3 rank scale, since it is more difficult even for human annotators to rank a phrase when the scale is more fine-grained (see Section 3.4 below).

In computational linguistics, most of collocation extraction methods use or are related to point-wise mutual information (PMI), which is one of the most popular ways to extract collocations or compounds. It was first proposed by (Church and Hanks, 1990) which used mutual information (MI) to extract word associations. The standard approach is to conceive the random variables of MI as lexical items, and approximate the probabilities of those random variables by counting lexical items in a corpus. So one can apply the concept of MI between lexical items x and y as follows:

$$PMI(x, y) = \log \frac{P(xy)}{P(x)P(y)} \tag{2.1}$$

where $P(x)$ is the probability of x in a corpus, and $P(xy)$ is the probability that x and y are consecutive within that corpus. But as they pointed out in their conclusion, the score took only distributional evidence into account. For example, score(set...for) is larger than score(set...down), of which the former is compositional while the latter is not and is more interesting. As explained in Chapter 3, one difference of our tightness measure is in how we use counting in a corpus to approximate the probabilities that define our measure, and its relationship to the lexical version of MI. In our case, the denominator is calculated by adding up all those non-adjacent occurrences where both x and y occur within the document (see details below). The intuition is that the evidence of two words appearing more closely will ensure the meaning being more related to the consecutive appearance.

(Lin, 1998) presented a method for non-compositional English phrase extraction based on the hypothesis that when a phrase is non-compositional, its mutual information differs significantly from the mutual information of phrases obtained by substituting one of the words in the phrase with a similar word. Phrases were represented in the form of triples: (**head type modifier**), which were extracted from a corpus automatically by a parser. In the triple, head and modifier were words in the input sentence and type was the type of the dependency relation. For example, one triple for the sentence "John married Peter" is (marry V : subj: N John), where "marry" is the head, "subj" is the type, and "John" is the modifier. He treated a collocation (head type modifier) as the conjunction of three events: ($\star$ type $\star$), (head $\star$ $\star$), and ($\star$ $\star$ modifier), where $\star$s represent other entities which are not in the collocation. The mutual information function for a triple is in the form of:

$$\lg \frac{P(A, B, C)}{P(B|A)P(C|A)P(A)} = \lg \frac{\frac{|\text{head type modifer}|}{|\star\star\star|}}{\frac{|\star\text{type}\star|}{|\star\star\star|} \frac{|\text{head type}\star|}{|\star\text{type}\star|} \frac{|\star\text{type modifer}|}{|\star\text{type}\star|}} \tag{2.2}$$

He assumed that modifier and head were independent to each other as long as type was settled. He set a range for mutual information of a phrase based on two assumptions. One assumption is that P(A,B,C) was normally distributed with mean $\frac{k}{n}$, where $k = |$head type modifer$|$, i.e. the frequency of the triple, and $n = |\star\star\star|$, i.e. the frequency of all triples, and $var = \frac{z\sqrt{k}}{n}$, where z is a constant related to confidence level. The other assumption is that the estimates of P(B|A), P(C|A), and P(A) were accurate. A collocation α is non-compositional if there does not exist another collocation β such that (1) β is created by substituting one of the words in the phrase α with a similar word and (2) there is an overlap between the 95% confidence interval of the mutual information values of α and β. He compared his results with two manually compiled English dictionaries: for the first, precision and recall was 15.7%, 13.7%; for the second, precision was 39.4%, and recall 20.9%. The unexpected difference of these results shows that even lexicographers can disagree about which phrases are non-compositional. The greatest contribution of the paper is that it combined linguistic information with statistical information to extract non-compositional compounds instead of just word associations like former work.

2.2 Chinese Word Extraction

Chinese is very different from alphabetical languages such as English, as there is no delimiter between Chinese words. So Chinese word segmentation is a premier step for most Chinese natural language processing (NLP) tasks, such as machine translation (MT) and information retrieval (IR). Many methods have been developed for Chinese word segmentation. There are dictionary-based methods and machine learning approaches. The common automatic segmentation errors can be classified into two categories: overlap errors and combination errors. *Overlap errors* occur when there is ambiguity of whether one character should be combined with either the former character or the latter character to form a word, e.g. for string "ABCD", whether to segment it as "AB|CD" or "ABC|D." Combination errors occur when there is ambiguity of whether two strings should be combined or not. For example, the sentence "佟大为|妻子|产下|一|女" (Tong Dawei's wife gave birth to a girl) can be segmented into "佟大|为|妻子|产下|一|女" (Tong Da gave birth to a girl for his wife), which is a combination error. One example of overlap error is segmenting "美和|服装" (Meihe Clothes) into "美|和服|装" (beautiful kimono). In the following, several recent Chinese segmentation methods will be introduced.

Dictionary-based approaches use a lexicon for segmentation. The most prevalent dictionary-based approach is the maximum matching method (Nie *et al.*, 2000). It segments a sentence either from left to right, or from right to left. First it finds the longest word in the lexicon that starts with the character which is the beginning of the sentence. Then remove this word, and start from the remaining sequence. If no word starts with the first character, then the character is taken as a word. For example, a sequence to be segmented is "ABCDE," and the lexicon is {AB, ABC, CD}. Then the segmentation is "ABC|D|E." First "ABC" is extracted as a word because it is the longest word

that begins with “A”. After “ABC” is taken, there is no match for “DE”, so it is assumed that one character is one word.

In (Feng *et al.*, 2004) a method based on statistical data called “context variety” is employed to extract candidates. The idea is to consider the variance of characters appearing on the right and left sides of a target character. Units with high variety are extracted, as such units appear in enough different environments to have the potential to be meaningful. For example, according to concordances “ABCDE”, “FBC”, “BCD”, the prefixes of “BC” will be (“A”, “F”, start of a sentence), and the suffixes are (“D”, end of a sentence). Then the variety value of “BC” is min(3,2), i.e. 2. They measured their extraction word list by comparing with a Chinese dictionary and calculated the precision. But this method, like many other Chinese word extraction methods, does not consider whether word units were compositional or not. For example, they extracted units such as “假冒伪劣商品” (fake and bad merchandise), which is not clearly a compositional word or a phrase.

(Xu *et al.*, 2006) is one of the few, as far as we know, which classify Chinese collocations according to their tightness. In this case tightness is distributed over 4 classes: idiomatic collocations, such as “‘缘木求鱼” (to climb a tree to catch a fish, meaning a fruitless effort); fixed collocations, such as “外交豁免权” (diplomatic immunity), in which two components can not be substituted by other words to carry the same meaning; strong collocations, such as “缔结同盟” (form alliance), with limited modifiability; and loose collocations, such as “合法收入” (lawful income), of which the replacement of components is not arbitrary. The input corpus, from which they extracted collocation candidates, is segmented and POS tagged. Their system was a pipeline combining a co-occurrence statistical model, a substitution method similar to the one mentioned above, and heuristic rules. They mainly combined two consecutive words, such as “缔结”(form) and “同盟” (alliance). They evaluated the extracted collocation precision according to a manually extracted set. One difference between our method and theirs is that our method locates Chinese semantic units in a continuous spectrum, while they classify them into 4 classes. Our method can be applied dynamically to meet different application needs, as generally it is difficult to separate between fixed collocations and strong collocations, between strong collocations or loose collocations. For example, “machine learning” may be a compound to some people, but may be a phrase to others. Another important difference is that their method is based on a large segmented and POS-tagged corpus, while our method is based on a large raw corpus. If the segmentation takes compositional words such as “假冒伪劣商品” (fake and bad merchandise) as units, then their method can not classify them correctly.

In the paper of (Qu *et al.*, 2008), an implemented pipeline was described, which included a basic step and several refinement steps, to extract Chinese nominal compounds from corpora. The definition of compound that they used was a polysyllabic noun that can be segmented into two or more morphemes either free or loan, i.e. whether a morpheme can represent a word alone or need to combine with other characters. The pipeline included four steps. In the first step, a method based on context variety (Feng *et al.*, 2004) was employed to extract candidates. Candidates with variety value

above a threshold were sent to the next step. The second step was a refinement post-segmentation step to filter names and numeric compounds. They used a forward-maximum algorithm to detect transliterated names such as "安吉丽娜茱利" (Angelina Jolie), and a lexicon of Chinese measure words for filtering numeric compounds such as "两百万个" (two million). In the third step, a syntactical step using tagging and parsing was employed to filter some false positives. At last, they employed an SVM classifier with mutual information and probability of POS-sequences as features for further uncovering pseudo-compounds. The test data was a set of random sample of 200 story texts from the Xinhua part of the Chinese Gigaword Corpus with compounds tagged manually. The recall and precision of their approach were measured. The result included different recall and precision for compounds of different lengths. Their method could achieve high precision, 98%, when the recall was 11% for 4-character compounds.

For machine learning Chinese word extraction approaches, one uses features such as part of speech (POS), gram frequency, position in sentences, contexts, etc. (Gao *et al.*, 2005) is one example of using abundance of resources. They developed a taxonomy that categorizes Chinese words into 5 classes: entries in a lexicon, morphologically derived words, factoids, named entities, and new words (words not in the former four classes). The lexicon is a combination of different dictionaries. Morphologically derived words are words such as "朋友们" (friend - plural) and "高高兴兴" (happily, reduplication of "高兴" (happy)). Factoids are time and date expressions. Named entities are frequently used Chinese names. New words are out of vocabulary words that are neither recognized as named entities, factoids nor derived by morphological rules. They chose candidate segmentations according to:

$$w^{\star} = \arg \max_{w \in GEN(s)} \sum_{d=0}^{D} \lambda_d f_d(w, s), \tag{2.3}$$

where s are character sequences, w are word class sequences generated by segmenting s, $f_d(w, s)$ are different features, and λ_d are weights of the features. There are 6 features, with the first one as $log(P(w))$ represented by a word class trigram model and others as $P(s|w)$ for every word class. In order to get $P(s|w)$, they constructed a finite state automata (FSA) for lexicon, factoids, and name entities class candidates extraction, while they constructed n-gram language models for the other classes. The linear weight vector λ_d was calculated using a gradient based method with manually segmented and annotated corpus as training data. The most interesting part is that they did not assume a fixed segmentation existed; instead they adjusted the segmentation after the aforementioned general model by an adaptation model. The adaptation model employed transformation rules which were extracted from the target corpus. The rules were in the form of:

Condition: word class

Actions: insert a new boundary between two component types or remove an existing boundary.

For example, if the target corpus is the U. Penn Chinese Treebank, first, the general model was employed to get a general segmentation; second, transformation rules were learned with the U. Penn

Chinese Treebank training data; finally, these rules were used for final segmentation on the test data.

One of the most popular Chinese word segmenters is ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System), a Chinese segmentation tool built by the Institute of Computing Technology, Chinese Academy of Sciences. It incorporates Chinese word segmentation, Part-Of-Speech tagging, disambiguation and unknown words recognition into a whole theoretical frame. The segmentation model is a class-based HMM model. Given a sequence A = $(a_1, ...a_n)$, let W = $(w_1, ...w_m)$ be the words sequence, C = $(c_1, ...c_m)$ be the corresponding class sequence of W, and W^* be the choice of word segmentation with the maximized probability. Then,

$W^* = \arg\max P(W|C)P(C)$,

$W^* = \arg\max \prod_{i=1}^{m} p(w_i|c_i)p(c_i|c_{i-1})$.

The classes are words in a lexicon, unlisted personal name, unlisted location name, unlisted numeric expression and so on. If w_i is in the lexicon, $c_i = w_i$, and $p(w_i|c_i) = 1$. Otherwise, $p(w_i|c_i)$ is probability that class c_i initially activates w_i. For the open source package, $W^* = \arg\max P(W) = \arg\max \prod_{i=1}^{m} p(w_i)$. $p(w_i)$ is obtained from the training corpus.

While the accuracy of automatic segmentation methods for homogenous data is above 95%, one problem arises: is it necessary to pursue higher segmentation accuracy? According to (Sproat *et al.*, 1996), the average agreement between human segmentor is 76%, so much less than this. As word segmentation is a pre-step for other NLP tasks, does better segmentation performance in the view of these gold standards get better results for IR and MT? Will it suffer from overfitting? We will describe some related work upon the effect of Chinese word segmentation for IR in the following.

2.3 Word Segmentation and Information Retrieval

The goal of information retrieval (IR) is, for a given query, find a set of documents which are most likely to be relevant to that query in a document collection. Chinese information retrieval (IR) systems employ the same framework as those of English. What differs Chinese IR and others is the parser, i.e. how to segment the query and the documents. Many approaches have analyzed the effects of Chinese word segmentation upon IR. The following are some of them.

Nie et al. compared six index construction methods (Nie, et al. 2000).

- 1. Use the longest matching with a small dictionary and with a large dictionary.
- 2. Combine the first method with unigrams.
- 3. Use full segmentation with or without adding unigrams, where full segmentation means extracting every word in a sentence. For example, if a dictionary is {“AB”, “ABCD”, “BC”, “D”, “E”}, for the sentence “ABCDE”, starting from “A”, words “AB”, “ABCD” will be extracted, and a new iteration begins with “B”.
- 4. Use overlap bigrams and unigrams;

- 5. Combine words with bigrams and unigrams.
- 6. Include an unknown word detection model.

Their experiments were based on TREC 5, 6 and their information retrieval system was SMART. The result showed that a larger dictionary had a slightly better IR performance; combining unigram with either words or bigrams was better, and unknown word detection helped. The authors suggested two reasons why bigram indexing did not suffer, despite many meaningless bigrams. One was that many meaningless bigrams were not in the queries. But they also pointed out a possible reason for not using bigram indexing in spite of its simplicity, i.e. it can not be used for cross-language IR, because the alignment of English words for semantically meaningless bigrams would diminish the result.

(Foo and Li, 2002) also analyzed the impact of different segmentation methods for IR. They compared the following methods: manual segmentation, pure bigram, overlap bigram, pure bigram with a one-character word list, and overlap bigram with a one-character word list. They showed that using the same segmentation method for queries and documents is superior to using different methods. The overlap bigram method got the best result. Their experiment was based on their own test collection, which contained 20 queries, and only 266 files, which made it possible for them to judge document relevancy exhaustedly. But the small size diminished the value of this research. Another contribution of this paper is that it analyzed several queries which distinguished IR systems to the largest extent. But it did not give sufficient detail of how bigram methods segmented queries or documents, and why some relevant documents were not retrieved based on this segmentation. In the related work part, they mentioned that one advantage of word segmentation method over bigram method was that better cross lingual results were obtained when the former was employed.

(Huang et al., 2000) investigated the effectiveness of different unit extraction methods and different term weighting methods in the context of Chinese IR. They classified unit extraction methods into character based and word based. They used longest match first for the word based extraction and single character, i.e. unigram for the character based extraction. They classified term weighting methods into two categories, single-unit weighting, and compound-unit weighting. Single-unit weighting was just normal BM25. Compound-unit weighting was a set of new weighting functions based on the observation,

$$w(t_1), w(t_2) < w(t_1 \wedge t_2) = w(t_1) + w(t_2) < w(t_1 \text{ adj } t_2) \tag{2.4}$$

which means the weight of a document containing only one term of a query is less than the weight of a document containing two terms of a query, and the latter is less than the weight of a document containing the two terms when they are adjacent (adj). In order to satisfy the right part of the equation, they proposed several compound weighting methods. Some of them are listed in Table 2.1.

Table 2.1: Compound unit weighting methods.

Weight Methods	$w(t_1 \text{ adj } t_2 \text{ adj } \ldots \text{ adj } t_j)$	$w(t_1 \wedge t_2 \wedge \ldots \wedge t_j)$
$Weight_1$	$\sum_{i=1}^{j} w_{t_i} + w_{t_1 t_2 \ldots t_j}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_2$	$\sum_{i=1}^{j} w_{t_i} + w_{t_1 t_2 \ldots t_j} + j^k$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_3$	$\sum_{i=1}^{j} w_{t_i} + w_{t_1 t_2 \ldots t_j} + \lg \frac{\sharp(t_1 \wedge t_2 \wedge \ldots \wedge t_j)}{\sharp(t_1 \text{ adj } t_2 \text{ adj } \ldots \text{ adj } t_j)}$	$\sum_{i=1}^{j} w_{t_i}$

In the table, $\sharp(t)$ indicates the number of documents containing the term t, k (k$\in [0, 2]$) and d are tuning constants. Adjacent $Weight_1$ is the sum of compound unit and component single units, $Weight_2$ has the strongest impact, while $Weight_3$ is a mild one, in which the last term is very similar to our tightness value function (in Section4.1). But we only consider $\sharp(r_1 \wedge r_2 \ldots \wedge r_c)$, where $r_1 \wedge r_2 \ldots \wedge r_c$ is the most reasonable segmentation of t_1 adj t_2 adj $\ldots$ adj t_j. For their experiments based on TREC 5, 6, it showed that character-based text processing performed better than word-based processing, and $Weight_3$ combined with BM26, an extended version of BM25, outperformed other score functions.

(Peng et al. 2002) suggested that the relationship between segmentation accuracy and retrieval performance was non-monotonic. Unlike previous methods, they did not take into account character-based methods, but only different word segmentation methods with various accuracies, from 44%-95%. They employed BM26, the function proposed by (Huang et al. 2000), and tuned parameters to simulate different IR systems. They evaluated segmentation performance on the Mandarin Chinese corpus, PH, compiled by Guo Jin (http://nora.hd.uib.no/corpora/1998-4/0205.html). The evaluation data for IR performance was also TREC 5, 6. For almost all the IR systems, the relationship between segmentation accuracy and retrieval performance was non-monotonic. In the analysis, they pointed out that "One possible explanation is that a weak word segmenter accidentally breaks compound words into smaller constituents, and this, surprisingly yields a beneficial effect for Chinese information retrieval."

It is worth mentioning the work of (Braschler and Ripplinger, 2004), which analyzed the effect of stemming and decompounding upon German text retrieval. Compounds are words built by concatenating several words. Similar to Chinese, Germanic languages, e.g. German, Dutch, Swedish, have many compounds. In their paper, they employed a system with no stemming or decompounding tokonization as the base line. They compared several stemming and compounding methods ranging from completely language-independent methods to components that use elaborate linguistic knowledge. The evaluation measure was average precision upon 90 queries created for Cross Language Evaluation Forum (CLEF) 2000 and 2001. The result showed a positive effect of stemming and decompounding for the German language. But they also pointed out that for some queries decompounding lead to negative results. So they suggested for further work, "how to automatically infer from corpus statistics how well compounds are represented by their constituents, and use this as a

factor in the decision of whether to apply compound splitting or not."

2.4 Information Retrieval Evaluation

IR performance evaluation is painful as there are many factors which will affect the result and one system may work effectively in some cases but poorly in others. The performance on a test collection is one popular way to evaluate and compare IR systems. There are three common components in a test collection: a query set, a document set, and a relevant document set. Constructing the standard relevant document set is critical for the test collection. Usually, it is impossible to extract all relevant documents as a serious corpus is too large. The well-known test collections, which were put together by NIST for the Test Retrieval Conferences (TREC), are gathered by the pooling method. The basic idea is that if a retrieval system is reasonably effective, the highest ranked documents will be excellent candidates for inclusion in the subset for judging. Every IR system submits the top 1000 retrieved documents (if retrieved documents are more than 1000.). Annotators judge the top 100 documents of every system. Then the result relevant document set will form the evaluation standard for different IR systems. Unseen documents are assumed to be not relevant.

Despite the popularity of TREC data, for some research areas it is not suitable, and specific test collections have to be created. This is also the case for our project. Unfortunately, even judging, for each query, the top 100 documents per IR system creates an unsustainable work load for small research organizations. This is why to evaluate IR systems based on the minimum test collection is a popular topic in IR.

In (Cormack *et al.*, 1998), they proposed two methods to reduce the effort for gathering relevant documents while still preserving the effectiveness of evaluation. One method was Interactive Searching and Judging. Four annotators operated separately. First, they sent a query to the MultiText project, their IR system for a specific topic. Second, they judged documents retrieved in the order in which they were returned until the frequency of relevant documents dropped to an extent that it seems no more relevant documents in the answer set can be found. Then they rephrased the query and repeated the process until the annotator decided it was covered sufficiently. The other method was Move-To-Front pooling, which was based on the top n documents each IR system retrieved. The difference with normal pooling methods is that the number of documents to be judged is not the same for each system. IR systems with better performance had more documents to be judged. The method determined its selection according to a priority queue, which selected the next document from the system that held the most recent relevant document. They evaluated these methods with four metrics against the relevant documents of TREC. The four metrics were, a simple count of the number of relevant documents in each collection, the root mean square of the differences in average precision (AP), the linear correlation between AP values, and the Kendall correlation (Kendall, 1955) between AP values. The result showed much similarity between results upon collections gathered by these two methods and the TREC data, while effort was reduced to one quarter of TREC

pooling method.

In (Carterette *et al.*, 2006), they proposed the Minimal Test Collection method to select documents to judge. Documents which can distinguish two IR systems to the largest extent in terms of AP were chosen. Recall the AP function (Equation 1.8), let x_i be the Boolean value indicating i's relevancy. Then Equation 1.8 can be rewritten as 2.5.

$$AP = \frac{1}{|R|} \sum_i x_i \frac{1}{rank(i)} \sum_{j \leq i} x_j \tag{2.5}$$

where j is a document ranked in front of i by system s, $rank(i)$ is the rank of document i. With arbitrary order of documents, the formula can be rewritten as 2.6.

$$AP = \frac{1}{|R|} \sum_i \sum_{j \leq i} x_i x_j a(i,j) \tag{2.6}$$

where $a(i,j) = \frac{1}{\max(rank(i), rank(j)}$. So the difference of two systems upon AP is $\triangle AP = \frac{1}{|R|} \sum_i \sum_{j \leq i} x_i x_j c(i,j)$, where $c(i,j) = \frac{1}{\max(rank(i,s_1), rank(j,s_1))} - \frac{1}{\max(rank(i,s_2), rank(j,s_2))}$, $rank(i, s_1)$ is the rank of document i by system s_1. If we can show that the sum of all positive $c(i,j)$ is larger than the sum of all negative c(i,j) , then $\triangle AP > 0$, and s_1 is better than s_2. Let D be the set of judged relevant documents, and D_u be the set of unjudged documents. It can be proven that by choosing the document $k = \arg\max_{k \in D}(w_k)$, $\triangle AP$ will get the greatest influence, where $w_k = \max(w_k^R, w_k^N)$, $w_k^R = \sum_j c(k,j) x_j$, and $w_k^N = \sum_j c(k,j) x_j$, where R means relevant, N means non-relevant.

In the experiment, they created 60 queries, and ranked the system in under three hours with 95% confidence, where a more specific of description of confidence will be given below.

In (Carterette 2007,), he made further research of *minimal test collection*, and proposed a way to measure the evaluation confidence based on the set. The confidence was defined as the probability that the difference of mean average precision was less than 0, i.e. $\triangle MAP < 0$, where mean average precision was the mean of average precisions for different queries. They showed that the mean average precision was normally distributed, so was the difference of mean average precision of two systems. In order to determine the confidence, the probability that an unjudged document was relevant had to be calculated first. They got this probability through three steps.

First, compute p(i, s, t), i.e. the probability that document i is relevant in topic t under system s, according to the document ranking. The higher the document ranks, the larger the probability is. And the more the relevant document is in the judged document set for topic t, the larger the probability is. So p is determined by the rank of the document in the system, and the topic. For example, for topic t, if system s_1 puts document A at rank 1, and system s_2 puts document B at rank 1, then p(A, s_1, t) = p(B, s_2, t).

Second, compute C(p(i, s, t)), i.e. the calibrated value of probability, in order to increase the probability under a good system over a bad system according to the judged documents.

Third, p(i, t) = f(C(I,s,t)), which is a logistic regression to combine all the systems' opinions. This is trained on the current judged documents.

The confidence shows the extent to which the current relevant judgments are enough to evaluate IR systems. If not, we can judge more documents. They used IR system results submitted to TREC to show that their confidence was robust, i.e. the MAP value was inside the confidence zone of the estimated value.

Chapter 3

Tightness Continuum Detection

Instead of classifying Chinese units into compositional or non-compositional, our tightness measure locates the tightness of units upon a continuum (Figure 1.1) based on the distribution of character sequence pattern frequencies. For a sequence of length n, there are 2^{n-1} potential segmentation candidates. In case of a 4-gram "ABCD," there are 8 candidates: Pt(ABCD), Pt(A|BCD), Pt(AB|CD), Pt(ABC|D), Pt(A|B|CD), Pt(A|BC|D), Pt(AB|C|D), and Pt(A|B|C|D). The intuition of our measure is that a frequent segmentation candidate will be a semantically reasonable segmentation, and the more frequent the pattern with the unified form is, the tighter a unit is. For example, in Figure 3.1, there are two examples of pattern distributions. The horizontal axis represents patterns, while the vertical axis represents the frequency of the patterns. On the left is the pattern distribution of the unit "耶路撒冷" (Jerusalem). The most frequent pattern is Pt(ABCD) which is the only correct segmentation for the unit. No other patterns for the unit could be found, an indication that it is very tight, i.e. non-compositional. On the right is the pattern distribution of the unit "城市建设" (city construction). The most frequent pattern is Pt(AB|CD), with Pt(ABCD) the second, and minor appearance of other patterns. It indicates that "城市建设" is compositional, which can be segmented into "城市" (city) and "建设" (construction).

The input of our method is the probability distribution of the sequences' patterns, i.e., potential segmentation candidates. The output is a continual tightness value, the greater the value, the tighter the unit. In the following section, we first give the description of counting pattern frequencies; then we will present the tightness measure. The examples and experiments target at 4-grams as 4-gram is the most common compound type in Chinese text, which is similar to the situation that bigram is the most common word type. This work was published in (Xu *et al.*, 2009).

3.1 Pattern frequency

Here we introduce how we acquire the input of the measure, i.e. pattern frequencies. As mentioned before, for a sequence of length n, there are 2^{n-1} potential segmentation candidates. In case of a 4-gram "ABCD", there are 8 candidates: Pt(ABCD), Pt(A|BCD), Pt(AB|CD), Pt(ABC|D),

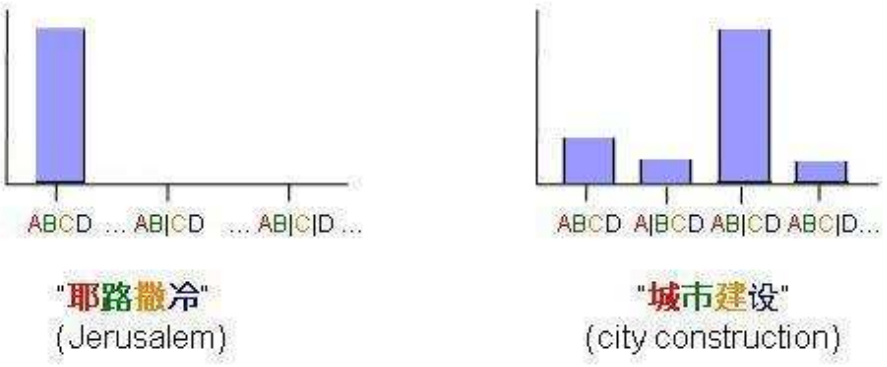

Figure 3.1: Two examples of pattern distribution

Pt(A|B|CD), Pt(A|BC|D), Pt(AB|C|D), and Pt(A|B|C|D). Each candidate is called a potential *pattern*. Note that typically only a subset of the patterns represent linguistically valid segmentations. We give a detailed description of patterns for the 4-gram "ABCD" below. First we introduce the component patterns for the 8 segmentation patterns of a 4-gram. In the following, the regular expression language is given in Java notation, and $\triangleq$ means "mark as." The * denotes characters other than "A", "B", "C", and "D."

- "[^A]BCD" $\triangleq$ Pt(BCD): a sequence with "BCD" without the character "A" in front of "BCD." Take the 4-gram "ABCD" as an example, the sequence "ABCD" and "CD**" do not match with this pattern, as "A" is in front of "BCD" for the first one and "B" is missing for the second one; while "**BCD" does match with this pattern.

- "ABC[^D]" $\triangleq$ Pt(ABC): a sequence with "ABC" without the character "D" following "ABC." Take the 4-gram "ABCD" as an example again, sequences "**ABCD" and "AB**" do not match with this pattern, while "ABC***" does.

- "AB[^C]" $\triangleq$ Pt(AB): similar to Pt(ABC). It is a sequence with "AB" without the character "C" following "AB."

- "[^B]CD" $\triangleq$ Pt(CD): similar to Pt(BCD). It is a sequence with "CD" without the character "B" in front of "CD."

- "A[^B]" $\triangleq$ Pt(A): similar to Pt(ABC). It is a sequence with "A" without the character "B" following "A."

- "[^A]B[^C]" $\triangleq$ Pt(B): a sequence with "B" without "A" in front of "B" and without "C" following "B." Take "ABCD" as an example again, "AB**" and "**BC*" do not match with this pattern, while "**B**" does.

- "[^B]C[^D]" $\triangleq$ Pt(C): is similar to Pt(B). It is a sequence with "C" without "B" in front of "C" and without "D" following "C."

- "[^C]D" $\triangleq$ Pt(D): is similar to Pt(BCD). It is a sequence with "D" without the character "C" in front of "D."

The former patterns are components of the 8 patterns of 4-grams. Following is the description of how the 8 patterns are counted.

- Pt(ABCD): if the whole unit appears in one document, then we say the document is evidence for this pattern and the frequency count of Pt(ABCD) is incremented by 1.

- Pt(A|BCD): if Pt(BCD) and Pt(A) are inside a document, then we say the document is evidence for this pattern and the count of Pt(A|BCD) is incremented by 1. Take the 4-gram "ABCD" as an example again, sequences "**ABCD" and "**BCD" do not match with this pattern, as "A" is in front of "BCD" for the first one and Pt(A) is missing for the second one; while "A****BCD" does match with this pattern.

- Pt(AB|CD): if Pt(AB) and Pt(CD) are inside a document, then we say the document is evidence for this pattern and the count of Pt(AB|CD) is incremented by 1.

- Pt(ABC|D): if Pt(ABC) and Pt(D) are inside a document, then we say the document is evidence for this pattern and the count of Pt(ABC|D) is incremented by 1.

- Pt(A|B|CD): if Pt(CD) is in a document and the document contains Pt(A) and Pt(B), then we say the document is evidence for this pattern and the count of Pt(A|B|CD) is incremented by 1. Take the 4-gram "ABCD" as an example again, sequences "A**CD" and "AB*CD" do not match with this pattern, as Pt(B) is missing for the first one and both Pt(A) and Pt(B) are missing for the second one; while "A****B*CD" does match with this pattern.

- Pt(A|BC|D): if a document contains Pt(BC), Pt(A), and Pt(D), then we say the document is evidence for this pattern and the count of Pt(A|BC|D) is incremented by 1.

- Pt(AB|C|D): similar to Pt(A|B|CD). It is a document contains Pt(AB), Pt(C) and Pt(D).

- Pt(A|B|C|D): if a document contains Pt(A), Pt(B), Pt(C), and Pt(D), then we say the document is evidence for this pattern and the count of Pt(A|B|C|D) is incremented by 1. Take the 4-gram "ABCD" as an example again, the sequence "**BA**C**D" matches this pattern.

Whenever one of the 8 segmentation patterns occurs in a document, this document is evidence for the pattern, and the frequency count of the pattern is incremented by 1. One document can be evidence of several patterns. For example, for 4-gram "ABCD", the sequence "ABCD****AB**CD" is evidence of Pt(ABCD) and Pt(AB|CD).

The pseudocode to extract these patterns' frequencies is given below (Table 3.1). In order to speed up counting frequencies, we build inverted indexes for the unigrams, bigrams, trigrams, and 4-grams in the respective corpus. For a 4-gram "ABCD", to get the frequencies of its 8 patterns, we first need to extract the document index of its 10 subsequence: "A," "B," "C," "D," "AB," "BC," "CD," "ABC," "BCD," and "ABCD." 10 pointers are set to the current position in the indexes. Of course, $\sharp Pt(ABCD)$ is the size of "ABCD"'s document list size. As the lists are sorted by document ID, we can iterate through all the document IDs. For every document, if the current position of a list is equal to it, then it means that a gram is in the document. If all the 4 unigram are in the document, calculate frequencies of all the other 7 patterns in the document according to the former patterns' definition; add the document frequency of patterns whose frequency in this document is greater than 0 by 1. Move the record pointers which are pointing to the current documents to the next one in the corresponding lists. Break the iteration if one of the unigram lists reaches the end.

3.2 A Tightness Measure

We assume a unit is tight with respect to a chosen corpus if when all component characters of the unit appear in a document, they always appear in one consecutive form, i.e. in the form of the unit. So the more frequent the whole pattern is compared to other patterns which separate the component characters, the tighter the unit is. Consider the 4-gram "ABCD" again, the more frequent the pattern Pt(ABCD) is compared to other 7 patterns, the tighter "ABCD" is. In contrast to pointwise mutual information, where frequency counts of parts x and y are based on their appearance in the whole corpus, our method only considers documents where x and y both appear. This is a better way to catch semantic realations between the compound and its parts. For example, if "machine" occurs in one document but not "learning", then that "machine" can be a car engine, a copy machine, other than a computer. Generally speaking, this is a way to introduce rudimentary word sense disambiguation which PMI ignores completely.

Instead of using patterns that separate a unit into more than two parts, we consider only patterns that segment a unit into two parts. One reason is the intuition that the greater order a gram is, i.e. the longer a gram is, the better it can hold specific semantic intention. A document with "医生" (doctor) and "护士" (nurse) will have a greater chance to be related to "医生护士" (doctor nurse) than a document with "医" (cure), "生" (born), "护" (protect), and "士" (person). Another motivation is that the patterns for which a 4-gram separates into two parts are also observations about the remaining three or four part segmentation candidates. For example, if the segmentation of a 4gram AB|C|D is semantically meaningful, then observations Pt(ABC|D) and Pt(AB|CD) are also possible; if a 4-gram can be segmented into A|B|C|D, then all the 8 pattern observations are possible. Take "我很想你" (I miss you very much) as an example. It can be segmented into "我(I) | 很(very much) | 想(miss) | 你(you)", so Pt(A|B|C|D), Pt(AB|C|D), Pt(A|B|CD) etc. might occur. For "机器学习" (machine learning), the semantically reasonable segmentation is "机器(machine)

Table 3.1: Pseudocode for calculating patterns' frequencies

Input: *index1* (inverted index of unigram), *index2* (inverted index of bigram),
index3 (inverted index of trigram), *index4* (inverted index of 4-gram),
and 4-gram "ABCD"
Process:
1. GET document index lists of 10 subsequences of "ABCD."
2. INIT *record*[10] to the start of the 10 lists
3. INIT $\sharp Pt[8]$ to zero
4. $\sharp Pt[0] =$ the size of the document index list of the sequence "ABCD"
5. FOR every document in the Gigaword
6. INIT *currentFreq*[10] to 0
7. FOR every record
8. IF *record[i].docID* = this document ID THEN
9. *currentFreq[i]* = *record[i].freq*
10. ELSE
11. *currentFreq[i]*=0
12. ENDIF
13. FOREND
14. IF all the unigrams' *currentFreq* >0 THEN
15. SET freqANB to curFreq of "A" - curFreq of "AB"
16. SET freqNABCD to curFreq of "BCD" - curFreq of "ABCD"
17. IF freqANB > 0 AND freqNABCD > 0 THEN
18. INCREMENT document frequency of Pt(A|BCD)
19. ENDIF
20. SET freqABNC to curFreq of "AB" - curFreq of "ABC"
21. SET freqNBCD to curFreq of "CD" - curFreq of "ABC"
22. IF freqABNC > 0 AND freqNBCD > 0 THEN
23. INCREMENT document frequency of Pt(AB|CD)
24. ENDIF
25. SET freqABCND to curFreq of "ABC" - curFreq of "ABCD"
26. SET freqNCD to curFreq of "D" - curFreq of "CD"
27. IF freqABCND > 0 AND freqNCD > 0 THEN
28. INCREMENT document frequency of Pt(ABC|D)
29. ENDIF
30. SET freqNABNC to curFreq of "B" -
curFreq of "AB" - curFreq"BC" + curFreq of "ABC"
31. IF freqANB > 0 AND freqNABNC > 0 AND freqNBCD > 0 THEN
32. INCREMENT document frequency of Pt(A|B|CD)
33. ENDIF
34. SET freqNABCND to curFreq of "BC"
- curFreq of "ABC" - curFreq of "BCD" + curFreq of "ABCD"
35. IF freqANB > 0 AND freqNABCND > 0 AND freqNCD > 0 THEN
36. INCREMENT document frequency of Pt(A|BC|D)
37. ENDIF
38. SET freqNBCND to curFreq of "C" -
curFreq of "BC" - curFreq of "CD" + curFreq of "BCD"
39. IF freqABNC > 0 AND freqNBCND > 0 AND freqNCD > 0 THEN
40. INCREMENT document frequency of Pt(AB|C|D)
41. ENDIF
42. IF freqANB > 0 AND freqNABNC > 0 AND freqNBCND > 0 AND freqNCD > 0 THEN
43. INCREMENT document frequency of Pt(A|B|C|D)
44. ENDIF
45. ENDIF
46. FOR every record[i]
47. IF *record[i].docID* = this document ID THEN
48. SET record[i] to the next record of the list
49. ENDIF
50. FOREND
51. IF one of lists A, B, C, D is at the end THEN
52. BREAK
53. ENDIF
54.FOREND
Output: the document frequency of 8 patterns $\sharp Pt[i]$

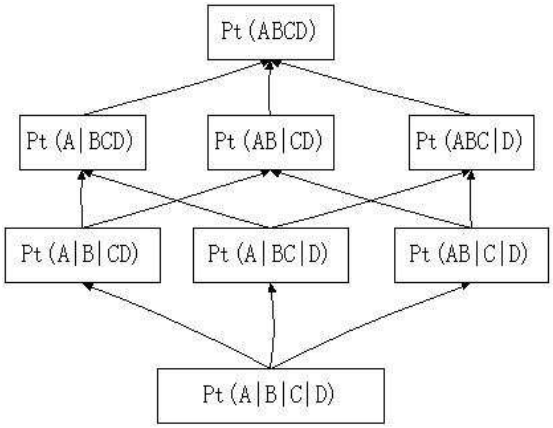

Figure 3.2: The lattice of the 8 patterns

|学习(learning)", Pt(A|B|C|D) will be rare as compared to Pt(AB|CD). Figure 3.2 shows the lattice of a 4grams' 8 patterns, which present the former idea. The link between two level patterns means when a lower level pattern is semantically meaningful, the related higher level patterns are likely to occur. The relation is transitive.

Among patterns that segment a unit into two parts, we assume the most frequent one is the most semantically reasonable one. For "机器学习", we expect Pt(机器|学习) will be more frequent than Pt(机|器学习) or Pt(机器学|习) ("机", "器", and "习" are bound morphemes which can not be a word alone).

With these observations, we propose the following tightness measure,

$$ratio = \begin{cases} \frac{\sharp Pt(\text{whole unit})}{\max(\sharp Pt(\text{patterns segmenting the unit into two parts}))+\frac{1}{N}} & \text{if } \sharp Pt(\text{whole unit}) > \sigma \\ \text{undef} & \text{otherwise} \end{cases} \tag{3.1}$$

where $\sharp$ means frequency, σ is a threshold to exclude rare patterns, which is set as 50 in the following experiment, and N is a smoothing factor which is set as the number of documents. When the first part of the denominator is zero, the ratio of the unit will be very high. This is reasonable, as we only calculate ratios for units whose frequency are greater than some threshold and we assume for those units it is the tightest when there is no evidence of patterns which separate the units. For 4-grams, the function is,

$$ratio = \begin{cases} \frac{\sharp Pt(ABCD)}{\max(\sharp Pt(A|BCD),\sharp Pt(AB|CD),\sharp Pt(ABC|D))+\frac{1}{N}} & \text{if } \sharp Pt(ABCD) > \sigma \\ \text{undef} & \text{otherwise} \end{cases} \tag{3.2}$$

This tightness measure can be used to compare tightness between units. Moreover, we can set a threshold and assume grams with a tightness value above the threshold as non-compositional when we extract Chinese semantic units.

3.3 Chinese Segmentation

There are two ways to employ the tightness measure for Chinese segmentation. One is revising the segmentation of some traditional segmenters, e.g. ICTCLAS. ICTCLAS is a segmentation model trained by an HMM model upon a segmented corpus. More details of the ICTCLAS segmenter have been introduced in Section 2.2. The other is directly segmenting the sentence according to 4-gram pattern distributions with some tightness thresholds. More specific descriptions are given below.

For the revising method, we regard the extracted 4-gram compounds as new entries of the dictionary. It is similar to the popular longest match first method, while the characters in the longest match first method turn into units of the ICTCLAS segmentation result and the dictionary is our list of compounds. The pseudocode of re-segmentation is given below (Table 3.2 and 3.3). For every Chinese character sequence chunk, the consecutive words are combined if the combination is in the dictionary. We do not consider words longer than 4, as the dictionary only has 4-grams. Let's take the ICTCLAS segmented sequence "菲律宾|皮|纳|图|博|火山|爆发" (Philippines's Mount Pinatubo eruption) as an example. First "菲律宾(Philippines) | 皮(skin)" will be checked, it is left unchanged as there is no combination word "菲律宾皮" in the dictionary. Then "皮(literal translation: skin) | 纳(literal translation: include) | 图(literal translation: picture) | 博(literal translation: large)" will be checked. Our method will recognize the proper noun "皮纳图博" (Minatubo) as it is tight and combine it. After that, "火山(volcano) | 爆发(eruption)" will be checked. It is left unchanged again as it is loose. So the revised segmentation will be "菲律宾(Philippines) | 皮纳图博(Pinatubo) | 火山(volcano) | 爆发(eruption)."

For the independent segmentation method, we set the additional thresholds σ_2, σ_3, and σ_4, and employ the following rules for segmentation (Rules 3.3). The intuition comes from the pattern lattice. For the patterns on the same level, the more frequent a pattern is, the more reasonable is that segmentation. Between two levels, the more frequent the upper level is, the tighter the pattern is.

$$
\begin{aligned}
&\text{if} \\
&\quad v_1 = \frac{\sharp Pt(ABCD)}{\max(\sharp Pt(A|BCD), \sharp Pt(AB|CD), \sharp Pt(ABC|D)) + \frac{1}{N}} > \sigma_2 \\
&\text{then "ABCD" is one unit;} \\
&\\
&\text{else if} \\
&\quad v_2 = \frac{\max(\sharp Pt(A|BCD), \sharp Pt(AB|CD), \sharp Pt(ABC|D)) + \frac{1}{N}}{\max(\sharp Pt(A|B|CD), \sharp Pt(A|BC|D), \sharp Pt(AB|C|D)) + \frac{1}{N}} > \sigma_3 \\
&\text{then "ABCD" is segmented into two parts;} \\
&\\
&\text{else if} \\
&\quad v_3 = \frac{\max(\sharp Pt(A|B|CD), \sharp Pt(A|BC|D), \sharp Pt(AB|C|D)) + \frac{1}{N}}{\sharp Pt(A|B|C|D) + \frac{1}{N}} > \sigma_4 \\
&\text{then "ABCD" is segmented into three parts;} \\
&\\
&\text{else} \\
&\text{"ABCD" is segmented into four parts;}
\end{aligned}
\tag{3.3}
$$

We use the former rules upon overlap 4-grams and a simple voting method to segment the whole corpus. Sequences with length less than 4 are segmented by the longest match first method with a

Table 3.2: Pseudocode for the revising segmentation algorithm

```
Segment a sentence:
Input: a sentence
Process:
1. INIT result seged to be empty
2. INIT a buffer s to be empty as the next chunk to be segmented
3. INIT length=0; (it is the length of characters except space in s)
4. FOR every word in the original sentence
5.          IF it is not a Chinese word THEN
6.               CALL segment the buffer s; (see table 3.3 for more specific description)
7.              re-INIT the buffer s and length
8.              CONTINUE
9.          ENDIF
10.          IF length+ word length>=4 THEN
11.               IF length EQUALS TO 0 THEN
12.                    attach the word and a space to seged
13.               ELSE
14.                    IF length+word length >4 THEN
15.                         trace back one word
16.                    ELSE
17.                         attached the word and a space to buffer s
18.                    ENDIF
19.                    result = CALL segment the buffer s
20.                    IF result contains only one part THEN
21.                         attach it to seged
22.                     ELSE
23.                         attach the first part to seged and trace back all other parts
24.                     ENDIF
25.                    re-INIT the buffer s and length
26.               ENDIF
27.          ELSE
28.                attach the word to buffer s, renew length
29.          ENDIF
30.FOREND
31.result = CALL segment the buffer s
32.attach result to seged
Output: segmented sentence seged
```

Table 3.3: Pseudocode for the revising segmentation algorithm, segmenting a sequence chunk

```
Segment the buffer s
Input: a compound list, the buffer s;
Process:
1. IF there is only one word in s THEN
2.          RETURN s
3. ENDIF
4. FOR every subsequence of s which starts from beginning of s (from longest to shortest)
5.          IF the subsequence is in the list THEN
6.               RETURN the subsequence + " " + the remaining part
7.          ENDIF
8. FOREND
```

dictionary. Otherwise, every 4-gram in the sequence is segmented by the former rules. For example, for the sentence "我今天迟到了" (I was late today), three 4-grams' patterns will be analyzed: "我今天迟," "今天迟到," and "天迟到了." If only one 4-gram contains an interval, the segmentation of that interval solely depends upon the 4-gram. If two 4-grams contain the interval, the segmentation of that interval depends upon the two 4-grams. If the two 4-grams do not agree upon the segmentation, a confidence value is calculated as in Equation 3.4,

$$confidence = v_i - \sigma_{i+1}, \tag{3.4}$$

where $i \in [1, 2, 3]$. If three 4-grams contain the interval, the voting result is employed to decide the segmentation. Take the sentence "我今天迟到了" (I was late today) as the example again. If the result of the first 4-gram segmentation is "我|今|天|迟," the second is "今天|迟到," and the third is "天|迟到|了." Then the interval between the character "我" (I) and "今" (today, a bound morpheme) is segmented. The interval between the character "今" and "天" (day) will be decided by the confidence between the first two segmentation patterns. If the confidence of the second segmentation is greater, then the segmentation is "今天," i.e. no delimiter is set in the second interval. The interval between the character "天" (day) and "迟" (late) is segmented according to the voting among the three 4-gram segmentations.

3.4 Experiments

To evaluate if our tightness measure does capture the tightness of strings, we conducted two experiments. First, we use our method to rank 300 4-gram Chinese strings, which include non-compositional words, compositional words, and phrases, according to their tightness. We then compare the result with a manually created gold standard ranking (the 300 phrases and their manual ranks are given in the Appendix). In the second experiment, we rank all the 4-grams in the Chinese Gigaword corpus according to their tightness, and assume the top 3,000 are non-compositional semantic units, such as idioms or transliterated names. We then compare these 3,000 grams with a dictionary. Note that our method is not only limited to 4-grams but, we take 4-grams as an example because 4-gram compounds are more prominent than others, just as character bi-grams are prominent for simple words in Chinese.

To evaluate the segmentation performance for our two segmentation methods, we segment documents of the *Chinese Treebank* (Xia *et al.*, 2000), a sgemented, part-of-speech tagged, and fully bracketed corpus that currently contains 500 thousand words (over 824K Chinese characters).

3.4.1 Rank Similarity

In this experiment, we compare the rank of 300 4-grams using our tightness measure and the PMI measure, across a variety of corpora. For our test a selection criterion was that all the 4-grams appear in both the Sogou query logs of March 2007 and the Chinese Treebank, and were tagged as NPs in

Table 3.4: Sample piece of Sogou query logs.

00:00:00 34217485189702995 [南粤双色球开奖结果] 3 1 www.0769888.com/qsc0769/849934712.html
...
00:00:04 34217485189702995 [南粤双色球开奖结果] 3 2 www.0769888.com/qsc0769/849934712.html
00:00:04 34062155775183716 [网易聊天室] 1 1 chat.163.com/
00:00:04 04324790273288531 [西安婚纱道具] 8 1 dzh.mop.com/topic/readSub_6280165_0_0.html
...
00:00:12 04324790273288531 [西安婚纱道具] 9 2 www.029apple.com/newforum/hAnnounceShow.asp?HFA_ID=30862&nCurpage=1
...
00:01:01 04324790273288531 [西安婚纱道具] 18 3 vip.wedchina.com/bbs/dispbbs.asp?boardID=41&ID=138380&page=1

the Chinese Treebank. The tightness of these phrases was measured over the statistical information in the following 6 corpora.

- 4 sets of snippets from 4 Chinese search engines, Baidu, Sogou, Google, and Yahoo!. We sent a query to the search engines for each of the 300 4-grams and recorded about 500 snippets for each query. For example, we sent "可口可乐" (Coca-Cola) to Baidu, record the first 500 snippets, and calculated the frequency of Pt(可口可乐), Pt(可|口可乐), etc., based on these 500 snippets.

- Web pages clicked for queries in the Sogou query logs where the 300 4-grams represented the full or a part of the users' queries (cf. Table 3.4 that shows a sample of the Sogou query logs. The respective documents have been downloaded for the experiment.) The first record in the table is for query [南粤双色球开奖结果] posed at 00:00:00 by user 34217485189702995. URL "www.0769888.com/qsc0769/849934712.html" ranks third by Sogou search engine for that query and is the first URL the user clicked for that query. If one of 300 phrases is "网易聊天' (Wangyi Chatting), then web page "chat.163.com/" will be considered as a support document for "网易聊天" because of the third query log in Table 3.4.

  To reduce the noise that comes with the webpages, we extracted content blocks that contain all characters of the query instead of extracting the whole page by "htmlparser." In the experiments, some URLs were no longer available.

- The simplified part of the Chinese Gigaword Corpus, which consists of 142 files with together 817348 documents, totaling approximately 0.5GB in compressed form. In order to get pattern distributions of 4-grams from the Chinese Gigaword corpus, we needed to extract documents that contain subsequences of 4-grams. So we build inverted indexes for unigrams,

bigrams, trigrams, and 4-grams in Gigaword using an open source Lucene package (Hatcher and Gospodnetic, 2004).

After filtering out 4-grams from our test set whose sum for the 8 pattern frequencies is less than 50 or the frequency of Pt(ABCD) is zero, 297 4-grams remained from Baidu, 266 4-grams remained from Sogou, 300 4-grams remained from Google, 295 4-grams remained from Yahoo!, 230 4-grams remained from the web pages, and 283 4-grams remained from Gigaword. We calculated the tightness value of 4-grams according to these different corpora and sorted them in descending order based on this value. So rank 1 was the tightest.

To evaluate the difference between our method and pointwise mutual information, we also ranked the 4-grams by pointwise mutual information according to the Chinese Gigaword corpus. To compute a 4-gram's mutual information, we segmented it into two parts according to the patterns' frequencies. For example, for a gram "ABCD," if $\max(\sharp Pt(A|BCD), \sharp Pt(AB|CD), \sharp Pt(ABC|D)) = \sharp Pt(A|BCD)$, then part1 = "A", part2 = "BCD". So the pointwise mutual information of a 4-gram is,

$$\log \frac{p(ABCD)}{p(part1)p(part2)} \tag{3.5}$$

where p means the probability.

To create a gold standard ranking, a human annotator (the author) ranked the 300 phrases of the test set on a 3 rank scale: rank 1 means very tight, for example, idioms or transliterated proper nouns, "澳大利亚" (Australia), "花花公子" (playboy); rank 2 means tight, such as compositional compounds, "人民银行" (people bank), "哈尔滨市" (Harbin city); and rank 3 denotes general phrases.

We use Kendall's τ to compare two rankings(Kendall, 1955):

$$\tau(r_a, r_b) = \frac{P - Q}{P + Q} \tag{3.6}$$

where P is number of same values between two rankings r_a and r_b, and Q is number of different values between two ranks. For comparison between automatic rankings, $P + Q = {}_nC_2$, where n is the size of intersection between two ranking domains. For example, for rankings based on Baidu and Google, there are 297 grams in the intersection. For comparison between an automatic ranking and the manual ranking, $P + Q = \frac{n_1 * n_2}{2} + \frac{n_1 * n_3}{2} + \frac{n_2 * n_3}{2}$, where n_i is number of 4-grams in rank i set. We do not compare grams in the same rank in this case as it is difficult to decide which is more tight, e.g., an idiom "一枝独秀" (outshine others), or an idiom "白手起家" (start from scratch).

For the τ measure, the following statistic, z, is approximately characterized by a standard normal distribution when the rankings are statistically independent:

$$z = \frac{3 * (P - Q)}{\sqrt{n(n-1)(2n+5)/2}} \tag{3.7}$$

Table 3.5: Rank similarities of measurements.

	Baidu	Google	Sogou	Yahoo!	Web pages	Gigawd_Ratio	*Gigawd_PMI*
Baidu	\	0.71	0.73	0.73	0.70	0.45	*0.39*
Google	0.71	\	0.74	0.74	0.72	0.45	*0.39*
Sogou	0.73	0.74	\	0.76	0.72	0.48	*0.41*
Yahoo!	0.73	0.74	0.76	\	0.74	0.48	*0.41*
Web pages	0.70	0.72	0.72	0.74	\	0.50	*0.43*
Gigawd_Ratio	0.45	0.45	0.48	0.48	0.50	\	*0.73*
Gigawd_MI	0.39	0.39	0.41	0.41	0.43	0.73	\
Manual rank	0.69	0.66	0.65	0.72	0.66	0.58	0.42

where n is the size of the list to be ranked, 300 in our case. Thus, if you want to test whether two rankings are statistically dependent, compute z, and find the cumulative probability for a standard normal distribution at $-|z|$. For a 2-tailed test, multiply that number by two and this gives you the p-value. If the p-value is below your acceptance level (typically 5%), you can reject the null hypothesis that the rankings are statistically independent and accept the hypothesis that they are dependent.

Table 3.5 shows the similarities of aforementioned tightness ranks against different corpora. "Baidu" means the ranking using our tightness measure against Baidu search engine snippets. "Gigawd_Ratio" means the ranking using our tightness measure against the Chinese Gigaword corpus. "Gigawd_PMI" means the ranking using the PMI measure against the Chinese Gigaword corpus. Because it is difficult to approximate the probability of components on the whole web corpus, we only calculate the PMI against the Gigaword corpus. The table shows Yahoo! as a corpus is closer to the manual rank than other corpora. The result is highly significant (when $\tau = 0.39$, $z = 10.04$, the probability of independency is < 0.00006).

The result shows more similarity between the automatic ranking using our approach and the manual ranking, as compared to the ranking using PMI, which means our method is a more reasonable measure of the tightness of Chinese units than PMI. For PMI, collocations such as "妇幼保健" (maternity and child care) are ranked as high, i.e. very tight, even higher than transliteration "马来西亚" (Malaysia); while our method ranks such collocations lower. As compared to other corpora, the Chinese Gigaword corpus gets the lowest similarity, which supports the insight of corpus linguistics: for statistical methods, a larger corpus leads to better results.

3.4.2 Non-compositional units extraction

In the second experiment, we ranked all 4-grams in the Chinese Gigaword corpus according to the tightness measure and analyzed the first 3,000 4-grams, which we assume as non-compositional Chinese semantic units, out of a total 830,809 4-grams. First we tried to find these 3,000 4grams in the "Modern Mandarin word dictionary." If a 4-gram is not in the dictionary (it is neither a lexical

item in the dictionary nor part of a lexical item), we query it on the Baidu search engine to check if it is some proper noun, such as a person name or a location name. If a 4-gram is in the dictionary or a non-compositional proper noun, for example proper nouns such as "上海市"(Shanghai city) are not non-compositional, but nouns like "上海" (Shanghai) are, then it is correctly extracted. The precision for the first 1000 4-grams is 94.3%; the precision for the first 2000 is 89.5%; and the precision for the first 3000 is 81.1%.

To compare with PMI, we rank all 4-grams according to the PMI measure. The precision for the first 1000 4-grams as non-compositional expressions is 66.3%. The PMI measure extracted more units that are loose collocations, or fixed but compositional expressions. Examples are expressions such as "虚报浮夸" (make a false report, exaggerate), "公道正派" (just, honest), and "叔叔阿姨" (uncle, aunt), which PMI ranked high while our measure ranked them low.

3.4.3 Chinese segmentation

In this experiment, we segmented the Chinese Treebank by the former two methods which employ the tightness measure. The results are compared with the manual segmentation of the corpus. Unlike other segmentation experiments which test their methods on the test set of the Chinese Treebank, we measured our methods on the whole Treebank corpus, as we use the Chinese Gigaword corpus to get the statistical information. We employed 10-cross validation in the experiment for result significance evaluation.

We compared our three segmentation methods to ICTCLAS. Tight_Combine is the ICTCLAS refined segmentation of using the 3500 non-compositional compound list from the Chinese Gigaword corpus in the former experiment. Tight_Split is the refined segmentation of Tight_Combine using Equation 3.3. Online_Tight is the segmentation using Equation 3.3 directly. Please refer to Section 3.3 for more specific description of these methods. For Tight_Split and Online_Tight, we set the thresholds σ_2, σ_3, and σ_4 specifically to 11, 0.01 and 0.01. The parameter σ_2 is set according to the observation that the percentage of non-compositional units is high when the tightness is greater than 11 for all the 4-grams in the Chinese Gigaword corpus. The other two parameters are set after trying several parameter pairs, such as (1,1), (1, 0.1), (0.1, 0.1), and (0.1, 0.01).

Table 3.6 shows the mean precision result over the 10 folders. The precision is the ratio of the number of correctly segmented intervals to the number of all intervals. The result shows our method improves the ICTCLAS segmentation result mildly. But the improvement is not significant. The only significant result is that Online_tight is worse than other methods. But the precision of Online_Tight is still satisfiable, as reported in (Sproat *et al.*, 1996) the agreement of Chinese segmentation between human annotators is merely 76%.

Surprisingly, there is a big gap between Tight_Split and Online_Tight, although they employ the same parameters. It turns out the major difference lies in function words. Since it is based on ICTCLAS, Tight_Split did a good job to segment function words such as verbal particles which rep-

Table 3.6: Segmentation precision upon the Chinese Treebank corpus.

ICTCLAS	88.8%
Tight_Combine	89.0%
Tight_Split	89.1%
Online_Tight	80.5%

resents past tense "了" and nominalizer "的." Online_Tight tends to combine these words with the consecutive one. For example, for "积累了" (cumulated), the Treebank and Tight_Split segmented it into "积累|了" (cumulate + particle); Online_Tight segmented it into "积累了." For "企业的" (company's), the Treebank and Tight_Split segmented it into "企业|的" (company + particle); Online_Tight segmented it into "企业的." Based on these observations, in the experiment of improving Chinese segmentation in IR (Ref(Chapter 4)), we employed Tight_Split and Tight_Combine instead of Online_Tight.

3.5 Discussion

One problem with our tightness measure is for words that are very frequent, the measured tightness is smaller than it should be. For example, according to our measure, "人民大学" (Renmin University) is much looser than "清华大学" (Qinghua University) as "人民" (Pronunciation: Renmin, meaning: people) is a common word. It is again the problem of word sense disambiguation. "人民" in the proper noun "人民大学" (Renmin University) is no longer the meaning of people. Although our method is better than pointwise mutual information as it filters out part of the problem, it still suffers when a word is very frequent. It is essential to add a weight p(the component's word sense in the unit | the component), i.e. the probability that when the component appears, the sense is the same as it appears in the unit.

3.6 Summary

Unlike previous research on Chinese word extraction, we believe Chinese semantic units fall into a continuum of connection tightness, ranging from very tight, non-compositional expressions, tight compositional words, phrases, and then to loose more or less arbitrary combinations of words. We proposed a tightness measure to locate Chinese semantic units within this continuum based on the statistical distribution of their component characters in a variety of corpora. Tightness ranks of 300 phrases computed by our measure based on different corpora, including search engine snippets, web pages, and the Chinese Gigaword corpus, showed high similarity with a human rank. A second experiment related to the extraction of non-compositional expressions showed promising results as compared to PMI when we evaluated the precision of our method against a dictionary.

Besides its value for linguistics, our approach can benefit applications such as machine trans-

lation and information retrieval. If a unit is non-compositional, the IR system should treat it as a word; if a unit is loose, it should be segmented. We can analyze IR performance employing different segmentations based on different thresholds of how tight a unit needs to be, to be considered as non-compositional. For example, there is no doubt that "月下老人" (match maker) is a single word and "上海哪有" (Shanghai where) are two words, but what about "机器学习" (machine learning), should it be segmented for IR or kept as a word? For the phrase-based machine translation (MT) method task, it is also important to decide whether a unit should be segmented or not. It is intuitive that IR tasks prefer a more fine-grained segmentation than MT tasks. Using our tightness measure, we can set a larger threshold for IR tasks and a smaller threshold for MT tasks to tune the granularity of word segmentation models. Such experiments will help us understanding the effects of Chinese word segmentation on NLP tasks. In the next Chapter, we will describe two methods we proposed to improve IR performance by this tightness measure.

Chapter 4

Applying The Tightness Continuum to Chinese Information Retrieval

Chinese information retrieval has the same framework as information retrieval of other languages. What differs is the word segmentation as there are no delimiters between Chinese characters in text. Queries need to be segmented into words. Documents need to be segmented before the inverted index construction. For example, the sentence "研究中文分词在信息检索中的作用" (Analyzing the effects of Chinese text segmentation upon information retrieval) is analogous to English text "AnalyzingtheeffectsofChinesetextsegmentationuponinformationretrieval." Word based indexing will segment the sentence into "研究(Analyzing) | 中文(Chinese) | 分词(segmentation) | 在(upon) | 信息(information) | 检索(retrieval) | 中(in) | 的('s) | 作用(effects)." As described in Chapter 1, two general methods exist to extract units from sentences for Chinese information retrieval: character-based and word-based. The current situation is that one character-based method, bigram, outperforms word-based methods. But how can a method which destructs the semantic meaning of units be better than word-based methods? Our intuition is that the current word-based Chinese information retrieval systems employ segmentation methods that are not wholly suitable for the task. Most of the segmentation methods are dictionary based methods, or use models trained by manually segmented data. They are not segmented according to the characteristics of IR systems. For example, a popular segmenter ICTCLAS, which is trained on a manually segmented corpus, keeps "科威特国" (Kuwait country) as a unit while for IR it is preferable to segment the unit into "科威"(Kuwait)|"特国"(country). We believe a word segmenter which can exploit semantics of Chinese units will benefit IR. So we propose two methods which imbed the former tightness measure into IR systems. While the first one, which tries to benefit IR performance by improved word segmentation, has a clear positive effect, especially on tight queries; the second method, which imbeds the tightness measure into the score function, did not show the effect of tightness measure on improved results. Nonetheless, the later method gives us some hint for tuning score functions to improve information retrieval. Through the experiments, we found that the current test collection data for Chinese IR evaluation is not suitable for testing the impact of semantic units on IR. Because of that, a special

test set available to the community has been created.

Three goals guided our investigation.

1. Stress the significance of proper treatment of semantic units for the performance of Chinese IR,

2. Document the lack of sufficient and appropriate data for Chinese IR evaluation, and

3. Examine procedures to integrate semantic tightness into Chinese IR systems

In what follows, we first present the description of the two methods. Section 4.2 then reports on the available data for Chinese IR evaluation and an approach to create new data. Section 4.3 presents results of our experiments on exploiting the tightness of semantic units for IR. Section 4.4 discusses the results. A short summary wraps up the Chapter.

4.1 Core Approach

We propose two methods to imbed the tightness continuum into Chinese IR. One is a segmentation method, which revises the output of a Chinese standard segmenter according to the tightness of units. Recall that we introduced this segmentation method in Section 3.3. The second method is an IR score function, which imbeds the tightness value into the IR score function. Note that these two methods can be combined or adopted separately.

(1) For the segmentation method, the intuition is that segmentation based on tightness of units will lead to improved IR performance. For example, keeping the tight 4-gram "皮纳图博" (transliteration for the volcano Pinatubo) as a unit should lead to better results than segmenting it into "皮(literal translation: skin)|纳(literal translation: include)|图(literal translation: picture)|博(literal translation: large)"; segmenting the compositional phrase "科威特国" (Kuwait country) into "科威特"(Kuwait)|"国"(country) will improve recall as compared to retrieve it as a unit. Figure 4.1 shows the framework of a Chinese IR system, which allows for the use of several possible segmentation methods. It includes character-based methods and word-based methods. Word-based methods include traditional segmentation models and our model. Our method takes the tightness measure and a segmentation result as input. It differs with traditional segmenters in that it takes semantic tightness into account. Moreover, we believe that individual units can have different tightness values within different corpora. For example, "机器学习" (machine learning) will be tighter in a computer science genre corpus than in a news genre corpus. So the best way to acquire statistical data for the tightness measure is using a corpus of the retrieval task's domain and genre. Our method provides the opportunity to use this information.

There are two steps to revise an initial segmentation: one is to combine components that should not have been separated, e.g., "皮 纳 图 博" (Pinatubo); the other is to split a sequence which is compositional, e.g., "科威特国" (Kuwait country). To combine components, we employ the

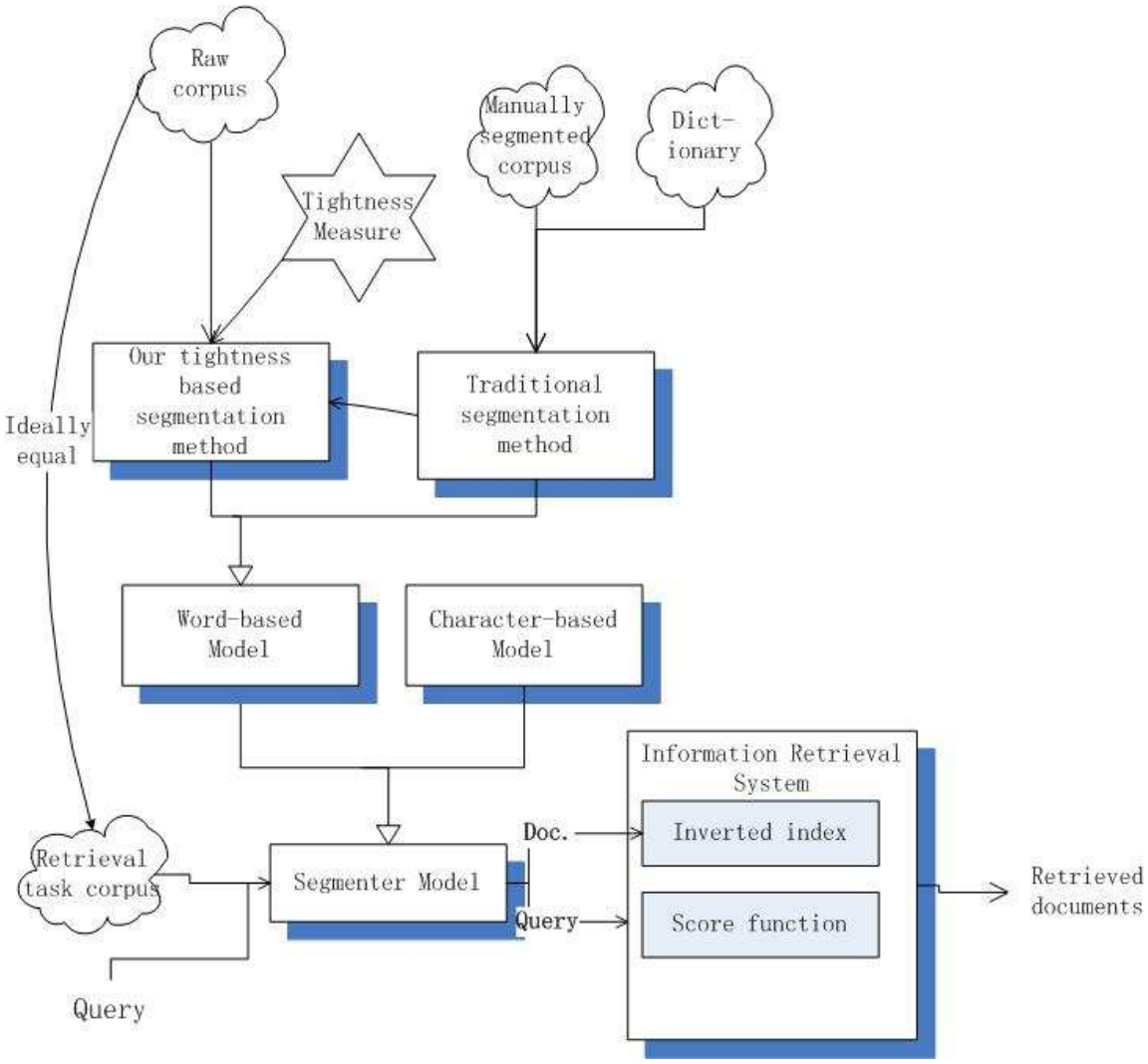

Figure 4.1: Segmenters in the framework of a Chinese IR system

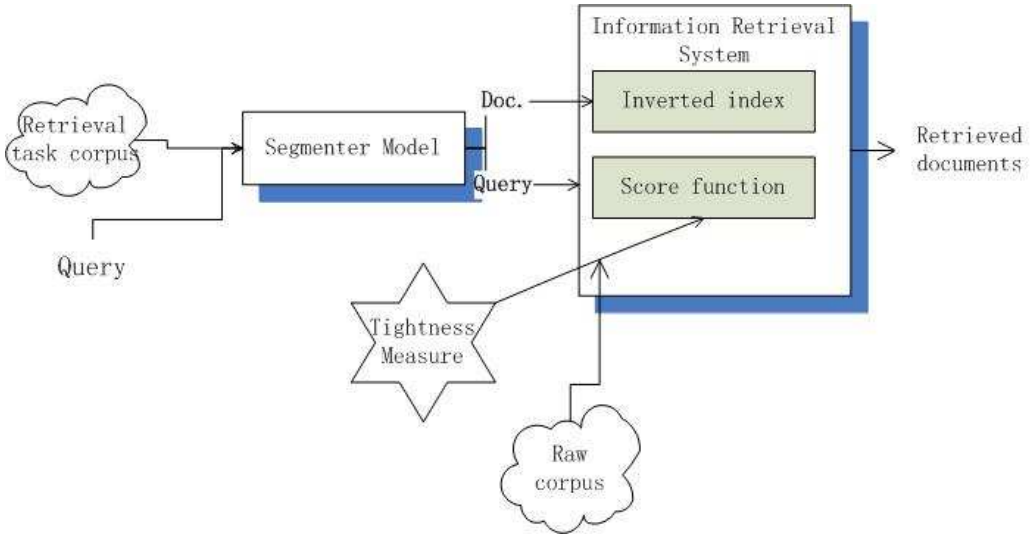

Figure 4.2: A dynamic score function in the framework of a Chinese IR system

Tight_Combine method in Section 3.4.3. To split a compositional units, we employ the Tight_Split method. We do not use the Online_Tight because of its problem of segmenting functional words. Refer to Section 3.4.3 and Section 3.3 for the description of these segmentation methods.

(2) Figure 4.2 shows the framework of imbedding the tightness measure into score functions in IR. The new score function method is based on the observation that one advantage of the overlap bigram indexing method is its bias towards documents which are more closely coordinated with the query, e.g., sharing more query terms, or having smaller term distance, i.e., proximity distance. Some lines of research suggest that the proximity distance measures on relevant documents will have smaller values than those of non-relevant documents (Tao and Zhai, 2009). Our hypothesis is that the coordination between terms in a query and terms in documents is more important for tight semantic units than for loose units. So we propose the adjusted score function 4.1, which takes the tightness value as a parameter.

$$Score(Q,D) = \begin{cases} \frac{T}{(1+Tight(Q))*N} \sum_{i=0}^{T} score(t_i, D) & \text{if } T < N \\ \sum_{i=0}^{N} score(t_i, D) & \text{if } T = N \end{cases} \tag{4.1}$$

where $score(t_i, D)$ is the score of term t_i in the document D. The function can employ different base score functions, such as BM25, tf⋆idf, or a probability language model. $Tight(Q)$ is the tightness value of query Q, N is the number of terms in the query, and T is the number of distinctive terms in both the document and the query. The function penalizes documents which do not contain all the query terms. The tighter a query is, the higher the penalty level. In the experiment, we employ BM25 as the base score function $score(t_i, D)$ (So we refer our function as BM25_Tight in the following).

4.2 Test Collection

After we analyzed the currently available Chinese test collections of TREC, we found that they are not suitable for analyzing the effects of Chinese segmentation on IR, in particular analyzing the advantages of appropriate segmentation of Chinese semantic units. One problem with TREC data is that the Chinese queries (topics) have too many keywords. Using ICTCLAS for segmentation, the average length of Chinese query is 12.2 words; while the average length of English ad-hoc queries in TREC-5 and 6 (English_topics 251-350) is 4.7. To remove language specific effects, we check TREC's English translation of the Chinese queries and still came to an avarage length of 7.2 as compared to the 4.7 for the English TREC. The problem with long queries is that, they introduce additional complicating effects that interact in ways difficult to understand. An example is the co-occurrence between different keywords in the base corpus. As most IR systems rely on bag-of-words indexing, the score functions can only estimate the importance of documents. Sometimes a correct segmentation fails because the score function gives a higher score to less important terms in a topic. For example, for query 47 (Trec-6 dataset), "菲律宾，皮纳图博火山，火山灰，岩浆，爆发" (Philippines, Mount Pinatubo, volcanic ash, magma, eruption), preserving the unit Pinatubo drops the average precision from 0.76 to 0.62 as compared to the segmentation "皮(literal translation: skin)|纳(literal translation: include)|图(literal translation: picture)|博(literal translation: large)." The top four correct documents for the split variant have ranks 1, 2, 3, and 4; whereas for the correct segmentation they rank 1, 2, 3, and 6 with a non-relevant document ranking the 4th. The non-relevant document is pd9108-1551, the four relevant documents are pd9106-1968, pd9106-2498, pd9107-3102, and pd9106-2372. The non-relevant document contains "岩浆" (magma), "火山" (volcano), "火山灰"(volcanic ash), and "爆发" (eruption); while the 4th relevant document contains the terms "菲律宾" (Philippines), "皮纳图博" (Pinatubo), "火山" (volcano) and "爆发" (eruption). When the unit "皮纳图博" (Pinatubo) is wrongly segmented, the score of the non-relevant document is 19, and the score of the relevant document is 22.7 with score(皮)=2.6, score(纳)=2, score(图)=1.8, and score(博)=2.6. When using the semantically reasonable segmentation, the score of the non-relevant document stays the same, while the score of the relevant document drops to 18.8 because the score of the unit is lower (score(皮纳图博)=5.3) than the sum of its 4 component scores. This is caused by the IR score function that discounts longer units. If, for example, the score of "岩浆" (magma) was smaller, the ranking result would be different.

Another problem with the TREC Chinese test collection is the small number of queries, 54, with an even smaller number of queries involving non-compositional words. When concerning the performance of IR with respect to different tightness values, the confidence of results based on this query set is not sufficient .

An alternative test collection, the *N2 Test Collection for IR systems* (NTCIR), is similar to TREC corpus. The documents are written in traditional Chinese. The project concentrates on Chinese Japanese Korean (CJK) languages. But its Chinese test collection also has only 50 queries, which

as argued is too small for our research.

A third potential set of Chinese IR evaluation data is the Sogou query log files. This data comprises queries and search results available as a web corpus. The problem is that this data set is not suitable in its raw form. It is a different kind of evaluation. The main weakness of this data set is that it is not relevancy annotated. It is impossible to judge recall and precision only from the documents clicked by the users. One another reason is that some relevant URLs are in the answer set not because of their literal semantic relation with a topic, but because of the implied semantics. For example, one query is "作你的爱人" (be your lover), which is a phrase, so for IR we should segment it into "作"(be) | "你"(you) | "的"('s) | "爱人"(lover). After we checked the relevant URLs, we found it is a movie's name, i.e. a proper noun, which should be kept as a word. Despite this, the Sogou data set proved to be very valuable in that it provides real world queries.

Because of this shortcomings of the available data sets, we created our own test collection, which comprises 200 queries with relevance judgements. The description of this test collection is given below.[1]

There are three components that define an IR test collection: a query set, a corpus from which relevant documents are retrieved, and relevance judgements for the documents in connection with the queries. In what follows, criteria of gathering these components are described.

First is the criteria for selecting the query set. For our purpose, the set of queries should contain both tight strings and loose strings, e.g, there should be tight strings such as "月下老人" (match maker), loose strings such as "上海海关" (Shanghai customs), and strings with tightness values in between, such as "机器学习" (machine learning). Furthermore, the queries should be reasonable, not artificially arising from introspection of the researcher. To meet these requirements we randomly choose 4gram noun phrases (tagged by ICTCLAS) from TREC, among which 50 were from a real data set, the Sogou query logs, and 150 were chosen manually based on the structure of these 50 queries.[2] Finally, the queries should not be too general (too many relevant documents found), nor too specific (no relevant documents). So we choose 4grams as queries whose document frequency in the TREC corpus is between 30 and 300.

The second set of criteria is about how to gather relevance judgements of documents with respect to the queries. We adopted the TREC Mandarin corpus as our retrieval corpus, which contains 24,959 documents. The size makes it impossible to judge every document. We also can not afford methods such as the pooling approaches employed by NIST to annotate the TREC collection for our query set. So we used the Minimum Test Collection (MTC) method (Carterette *et al.*, 2006). It pools documents in an order that documents which distinguish different IR systems to the largest extent will be judged first. We processed this method on a document set which contains all of the top 100 results of 8 IR systems (two score functions, tf⋆idf and BM25, 4 segmentation methods,

[1]For research purposes the queries and the TREC code numbers of the judged documents are available at http://cs.ualberta.ca/ yx2/IRTestCorpus.

[2]The query data can be downloaded at http://www.sogou.com/labs/dl/q.html.

unigram, bigram, ICTCLAS segmentation, and our Tight_Combine segmentation). The systems were implemented with the Lucene framework (http://lucene.apache.org/).

Last but, equally important, is the question of how to set the criteria for a document relevance judgement. Different annotators have different opinions about whether a document is relevant to a topic. Is having the query in a document sufficient as the criterion of relevance? A query, "Beijing airport," for example, may appear in many documents, of which the topic may or may not be related to the query. If a document contains the sentence "Chairman Mao arrived at the Beijing airport yesterday," of which the topic is about Chairman Mao but not the airport, should the document be considered as relevant? Concerning our goal, i.e., to analyze the relationship between Chinese word segmentation, especially compound segmentation, and IR, we use weak relevance judgments. For the given example this means it will be judged as relevant for the query "Beijing airport," We choose the weak relevance judgements as it is more related to score functions than word segmentation methods to distinguish weak relevance from strong relevance, that is, in how far the query is related to the topic of the document.

To summarize, our own test collection has 200 queries, and 100 judged documents per query with the TREC corpus as our base corpus. See the Appendix B for a more specific description of the system to gather the test collection.

4.3 Experiments

To measure the tightness of Chinese semantic units, pattern distributions of every 4-gram are extracted from the Chinese Gigaword corpus, which is larger and at the same time homogeneous with TREC in that both corpora contain newswire text. In case a 4-gram is not found in Gigaword, the distributions are extracted from the TREC corpus as a fallback solution.

For the method Tight_Combine, the threshold σ_1 is set as 11. For the method Tight_Split, we set the thresholds σ_2, σ_3, and σ_4 specifically to 11, 0.01 and 0.01. Parameters are set based on the result of segmentation upon the Chinese Treebank (cf. Section 3.4.3). Table 4.1 shows the result of comparing these two methods of word based indexing, with the word based indexing determined according to the standard segmentation ICTCLAS. Additionally the two character based indexing methods, unigram and bigram are documented. The performance of IR is measured by average precision (Equation 1.8). The results show that for word based indexing our method Tight_Combine is superior to the original ICTCLAS segmentation, while the result is not clear for Tight_Split. The tf*idf function shows better performance than BM25, which contradicts the suggestions of related work. After a deeper analysis, we found tf*idf in the employed IR system, Lucene, already takes the number of query terms in documents into concern, which is similar to the score function 4.1 when taking $Tight(Q) = 0$ (Function 4.2). It penalizes documents which do not contain all the query

Table 4.1: Result of IR systems with different segmenters

	tf⋆idf	BM25
ICTCLAS	68.89%	62.78%
Tight_Combine	69.83%	65.92%
Tight_Split	68.78%	63.40%
unigram	39.13%	33.59%
bigram	80.48%	71.81%

Table 4.2: Result of IR systems with different score functions

	BM25_Tight	BM25_Beta
ICTCLAS	70.83%	70.79%
Tight_Combine	71.23%	71.19%
Tight_Split	70.87%	70.95%
unigram	43.39%	43.25%
bigram	**81.25%**	81.15%

terms, but the penalty is not related to tightness.

$$Score(Q, D) = \frac{T}{N} \sum_{i=0}^{T} score(t_i, D) \tag{4.2}$$

The non-intuitive predominance of the bigram method against all word based indexing methods is related to the undersized corpus, and will be further discussed in Section 4.4.

To show that the improvement of the new score function is related to tightness value of strings, we compare it with using a constant value $\beta = 10$, instead of $Tight(Q)$, which refers to BM25_Beta in the following. The result is shown in Table 4.2. Comparing with Table 4.1, we found the tightness value seems to have the same effect with a constant value (we also tried constant 3, but the result is still similar). The reason for this behavior is discussed in the next section.

To give a more specific analysis of the methods for word segmentation with respect to the targeted phenomenon of semantic units, we classified the 200 queries into four categories according to their tightness as measured by function 3.1. The four classes are queries with tightness in ranges [$+\infty$, 10), [10,5), [5, 1), and [1, 0), which contain 54, 10, 31, 107 queries respectively. Queries in the range [$+\infty$, 10) are tight queries, such as "弗吉尼亚" (Virginia). Queries in the range [1, 0) are loose queries, such as "广告公司" (advertising company). Other queries are those compounds have ambiguous segmentations, such as "连锁反应" (chain reaction) and "红十字会" (Red Cross). Because the classification is based on our tightness measure, there are some errors. For example, "人民大学" is classified into loose queries while it should at least be in the middle range. But we think it is more objective than employing human annotators. The 4 classes covered the whole tightness continuum, i.e. the whole possible query set. So this classification will not bias towards our IR systems. Table 4.3 shows the average precision results with respect to these classes for the dif-

ferent word segmentation methods. For queries with tightness less than 10, the results of ICTCLAS and Tight_Combine are approximately equal, which is not surprising since with few exceptions they have the same segmentation for both queries and documents.

For the interesting case of segmentation of tight units, i.e. in the range [$+\infty$, 10), the results show clear superiority for IR systems based on the improved segmentation. Average precision is 86.44% for Tight_Combine as compared to 74.48% for standard word segmentation when using BM25. The advantage of Tight_Combine over ICTCLAS is that it recognized units such as "三来一补" (the three-processing and one compensation, which is an economic term). The ICTCLAS model segmented the term into "三" (three) "来" (come) "一" (one) "补" (compensation) as "三来一补" is an unknown word against ICTCLAS's training corpus. Besides, Tight_Combine combined units such as "平板玻璃" (plate glass) as the term is tight, while ICTCLAS segmented that unit into "平板" (plate) and "玻璃" (glass). This is evidence that word segmentation models based on tight measure is better than models trained on a human annotated corpus. Interestingly, Tight_Split is superior for word segmentation in the range [$+\infty$, 10), given that the segmentation for these queries is the same with Tight_Combine. When we analyzed the instances, we found it improves IR results of proper nouns. One possible explanation is that split of some proper nouns such as "弗吉尼亚州" (Virginia state) in documents improved the recall even when the segmentation of the queries stayed the same. For example, for query "弗吉尼亚" (Virginia), documents which contain "弗吉尼亚州" (Virginia state) should be retrieved. But as ICTCLAS treat "弗吉尼亚州" as a word, those documents are lost. Recall that Tight_Combine only combines units that are tightly connected, but not split units that are not tight. So Tight_Combine has the same act with ICTCLAS for these documents. Tight_Split segments the sequence into "弗吉尼亚|州" according to the Equation 3.3 and retrieved those documents.

In the range of [10, 5), the result is mixed. Tight_Split is worse than Tight_Combine and ICTCLAS as it segmented queries such as "连锁反应" (chain reaction) (Its tight value is 8.00). But it is better than Tight_Combine and ICTCLAS when it segmented queries such as "国际象棋" (translation: chess; literal translation: international chess (comparing to "中国象棋" (Chinese chess))) (Its tight value is 5.50). The result suggests the threshold of non-compositional terms should be less than 10.

In the range of [5, 1) and [1, 0), the result is also mixed. One reason for the low performance of Tight_Split is that the tightness measure is not precise for those queries, which affects the segmentation. For example, the split of queries "工人运动" (labor movement) (Its tight value is 0.27) and "中山大学" (Zhongshan University) (Its tight value is 0.84) harm the IR performance dramatically. Based on this observation, we think future work which segments queries manually according to their tightness will be necessary. If the manual segmentation is superior, then the hypothesis that segmentation based on tightness is better will be proved.

The difference between BM25 and BM25_Beta in the ranges [10, 5) and [5, 1) suggests that for

Table 4.3: Result of 4 categories

	[+∞, 10)	[10,5)	[5, 1)	[1, 0)
BM25				
ICTCLAS	74.48%	57.89%	61.05%	57.87%
Tight_Combine	86.44%	58.36%	61.25%	57.70%
Tight_Split	88.86%	51.16%	58.59%	53.17%
BM25_Beta				
ICTCLAS	84.60%	81.29%	69.75%	63.28%
Tight_Combine	86.44%	81.51%	69.86%	63.07%
Tight_Split	88.86%	82.48%	72.32%	60.39%

Chinese IR, it is better to segment text in a more fine-grained way, and combine terms through a score function. For example, for queries such as "连锁反应" (chain reaction), for which splitting the unit is worse, BM25_Beta decreases the negative effect of splitting dramatically. For the query "人寿保险" (life insurance) (Its tight value is 3.56), when using BM25, Tight_Split is worse than ICTCLAS (average precision 0.59 vs. 0.66); when using BM25_Beta, it is better than ICTCLAS (average precision 0.72 vs. 0.66).

4.4 Discussion

Our experiments showed that as one result, segmentation based on the tightness measure does improve IR performance for word based indexing when queries are tight. The preservation of tight semantic units and the split of loose phrases allows the system to retrieve relevant documents that are lost otherwise, and it improves the overall ranking of documents.

But the second result, the improvement of our score function seems to be related to the tightness in a more opaque way. Recall our hypothesis that, when a query is tight, the difference between relevant documents and irrelevant documents w.r.t. the number of query terms is more significant than when it is loose. We analyzed the judged documents to verify this hypothesis. For every query, we calculated the mean value of the difference between the number of query terms present in relevant documents and that present in irrelevant documents. Figure 4.3 shows a trend that the queries' tightness and the computed mean value decrease conjointly. But the trend is not stable. Furthermore, after we analyzed the retrieved documents, we found it is true that proximity distance measures on relevant documents have smaller values than those on non-relevant documents, but it is not related to query tightness. Nevertheless, this dynamic function may be helpful especially for long queries. When a query is long, some query terms might be semantically connected tighter than others. In this case, their coordination is more important than for the non-connected terms. The first query in TREC '最惠国待遇，中国，人权，经济制裁，分离，脱钩" (most-favored nation status, human rights in China, economic sanctions, separate, untie) after segmentation is written as "最惠国" (most-favored nation), "待遇" (status), "中国" (China), "人权" (human rights), "经济"

Table 4.4: IR result of 5 particular 4-grams

	tf⋆idf	BM25
unigram	31.79%	27.08%
bigram	64.49%	58.93%
ICTCLAS	74.80%	65.89%

(economic), “制裁” (sanctions), “分离” (separate), “脱钩” (untie). “经济” (economic) and “制裁” (sanctions) will be tighter than “经济”(economic) and “待遇” (status). We can cluster these terms into categories, while terms inside the category are combined by function BM25_Tight, terms between categories are combined by function BM25_Beta.

A result that has not been adequately addressed in the literature on word based segmentation is that the overlap bigram method, on the data available, always has the best retrieval performance. One popular explanation is that bigram is the most frequent word type in Chinese. We believe that besides the former reason, it is also related to the situation that the TREC corpus is too small to demonstrate the weakness of the bigram method, i.e., there are not enough negative documents for the bigram method in the TREC data. As an example, for 4-gram “机器学习” (machine learning), the bigram “器学” is rare, so the negative effect of this term as a false positive is neglectable. As a counter example, for 4-gram “美和服饰” (Meihe: a company name, clothes), the bigram “和服” (kimono) is frequent, and our hypothesis is that the performance of the bigram method performance will be inferior for queries of this kind. So we extracted five 4-gram queries according to the following criteria: 1. the most frequent pattern is Pt(AB|CD); 2. the number of documents is large enough, in which “BC”’s frequency is greater than 2 and there is no “AB,” “CD,” or “ABCD.” Note these 4-grams also have to satisfy our criteria of choosing queries of course. The 5 queries are, “上海南浦” (Shanghai Nanpu, a location name), “柏林市长” (Berlin Mayor), “韩国名将” (Korea famous athlete), “刑事处分” (criminal sanction), and “粮食品种” (food category). The result is given in Table 4.4. We still use the two basic score functions. There are only three segmentation methods since ICTCLAS and our segmentation methods have the same segmentation result for these 5 queries. This small test data does support our hypothesis, i.e., bigram segmenter model underperforms in comparison to word segmentation methods for these kind of queries.

To summarize, the current test collection of Chinese IR is not enough, whether in terms of corpus or the size of query set. It is essential to create more data for Chinese IR system evaluation in order to better analyze the advantage of different IR systems.

4.5 Summary

In this chapter, we have speculated how to treat Chinese semantic units to improve IR performance. Two methods have been proposed. One improves IR performance by employing Chinese segmenter

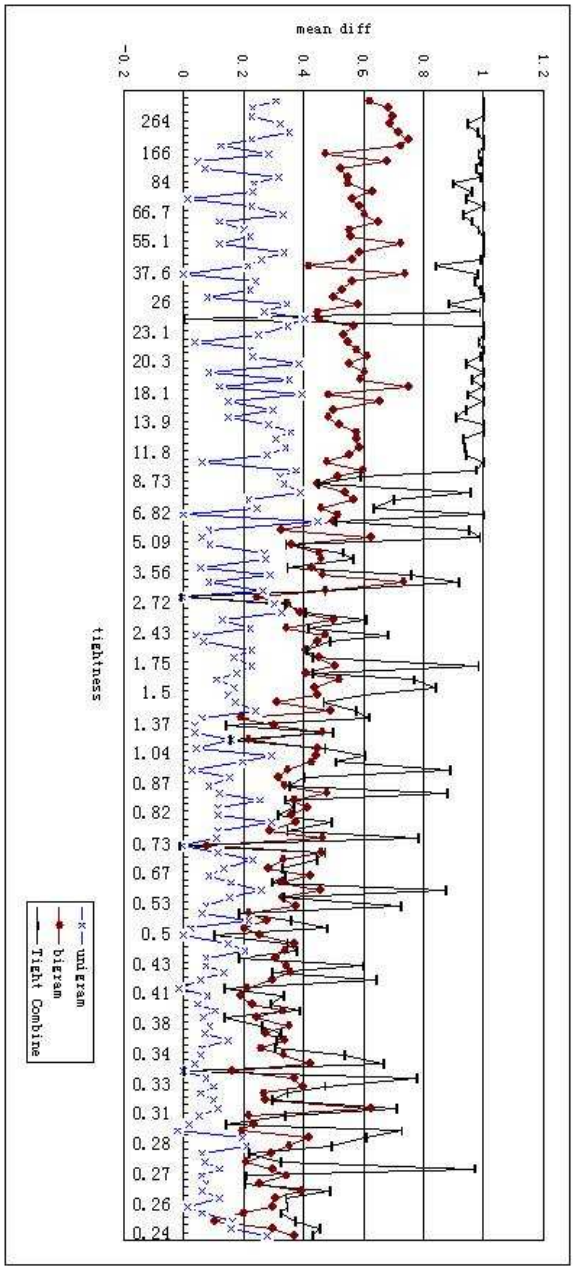

Figure 4.3: Difference between number of query terms in relevant documents and that in irrelevant documents

models which are based on a tightness measure. The other improves IR performance by imbedding the tightness measure into a score function. In our experiments, we have shown that average precision of IR systems using our segmentation method was higher than using standard segmentation for tight queries. The mixed result for queries in the range of [10, 0) suggests that future work which segments queries manually according to queries tightness will be necessary. If the manual segmentation is superior, then the hypothesis that segmentation based on tightness is better will be proved. Besides, in the future, it will be important to gather more queries and more judged documents to further analyze the effects of the adequate treatment of semantic units in Chinese information retrieval. We hope that we have created awareness of the shortcomings of presently available test sets, and the requirement of a relevance annotated query set related to a background corpus significantly larger than TREC. With these extended data, it will be interesting to evaluate an integration of the tightness measure into score functions, such as BM25_Tight.

Chapter 5

Conclusion

In the following two sections, we first present a summary of the dissertation. Then we will discuss open questions and future work.

5.1 Summary

The definition of a single word in written languages is always debatable. Word separators are common in modern orthography of languages using alphabetic scripts, such as English. But even in English, words may contain spaces, such as the word "kick the bucket." Chinese texts systematically do not contain delimiters between words. Talmy Givon said, "Today's morphology is yesterday's syntax." The boundary of words evolves over time. We believe that there are no clear word boundaries, whether in English or in Chinese. For example, in English, "going Dutch" is a word; "last year" are two words; but it is difficult to decide if "machine learning" is one word or two. In Chinese, "月下老人" (match maker) is one word; "上海哪有" (where in Shanghai) are two words. However, we are not sure about "机器学习" (machine learning). Therefore, we proposed the hypothesis that a Chinese semantic unit does not fall clearly into the binary classes of compositional or non-compositional, but into a continuum of tightness and looseness. We developed a measure based on the statistical pattern distribution of strings to locate the strings within the tightness continuum. The idea is similar to pointwise mutual information (PMI) in that two units appearing more often together are tighter. It differs from PMI in that instead of considering all instances of a term, we only count that cases when on document level it occurs with another unit, which we believe is an indirect approach of word sense disambiguation. Two experiments, tightness ranking of 300 phrases and non-compositional 4-gram extraction, show our measure does capture the idea of the tightness of Chinese units.

We believe this tightness measure can benefit many NLP tasks, such as machine translation, text classification, and information retrieval. For example, for IR, keeping the transliteration "皮纳图博" (Pinatubo) as a word is better than segmenting it into four words, as the unit is tight; segmenting "科威特国" (Kuwait country) into "科威特"(Kuwait) "国" (country) is better than keeping it as

one word, as the unit is compositional. So we embedded our tightness measure into IR systems to improve IR performance. Two methods were proposed, one is to segment queries and corpus according to the tightness measure and then retrieve documents according to this segmentation; the other is to embed the tightness value into a score function based on the hypothesis that coordination between terms in a query and terms in documents is more important for tight semantic units than for loose units. In order to analyze the effect of our methods, we created a test collection which contains 200 queries, since we found the current Chinese IR test collections are not suitable for analyzing the effect of Chinese compound segmentations upon IR. The experiments showed that the segmentation method employing our tightness measure improved the IR performance for tight queries. But the second method produced little improvement. The reason was that the importance of coordination between terms in the query and terms in documents was not related to tightness, at least for short queries which contain at most two terms.

5.2 Future Work

With our work we have shown the importance of further achievements in statistical semantic analysis of queries and index terms for word based Chinese in IR systems. More work needs to be done on two levels: improving the proposed function to measure the semantical tightness of potential units, and secondly analyzing word based Chinese IR in principal. Furthermore, our proposed measure can be employed on other languages with compounding phenomena. Following are some of our ideas.

As previously mentioned, one advantage of our method over MI is that we included an implicit variant of word sense disambiguation. But there are problems such as “人民大学” (Renmin University) which is much looser than “清华大学” (Qinghua University) according to our measure, where from a linguistic point of view both should have similar tightness. The reason is that “人民” (Pronunciation: Renmin, meaning: people) is a common word. But “人民” as in “人民大学” (Renmin University) is already not used in its original sense as people. So one promising direction of further research might be using word sense disambiguation to improve our tightness measure. Take “人民大学” (Renmin University) as an example again, other “人民” tokens in the text should not be taken into account.

As previously mentioned (Ref. Section 4.1), individual strings can have different tightness values within different corpora. For example, “机器学习” (machine learning) will be more tight in computer science genre corpus than in news genre one. Then one problem is, for balanced corpus, i.e. computer science genre corpus mixed with other genres, how can we analyze the tightness of “machine learning” appropriately? One example would be the mechanical terminology “放大器盘” (amplifier panel), for which we found all search engines (Baidu, Google, Sogou) returned bad results, e.g. webs about amplifiers. If we can figure out this terminology belongs to mechanical corpus and it is a tight compound, then we can improve the result.

The tightness fluctuation between different genres of corpora creates the noise for tightness measure in balanced corpus, but it can benefit terminology extraction. We can extract all the n-grams in different corpora, for example, the English Gigaword corpus and the Medline corpus. Then compare the fluctuation of tightness of n-grams between these two corpora. The more the fluctuation, the more likely it is a genre-specific term. For example, the tightness fluctuation of "night blindness" will be larger than that of "match maker" between those two corpus.

In another possible direction, we can extend our tightness measure for decompounding of languages such as German, in which compounds are very common. Recall the work (Braschler and Ripplinger, 2004), which analyzed the effect of stemming and decompounding upon German text retrieval. They decompounded all the words that are composed of more than one unit, such as "machine learning." They pointed out that some queries got worse results when using decompounding, based on which they suggested further work, "how to automatically infer from corpus statistics how well compounds are represented by their constituents, and use this as a factor in the decision of whether to apply compound splitting or not."

In terms of Chinese information retrieval, we think the current test sets are not big enough. We need more test queries related to a bigger corpus. Recall the performance of the bigram method, which we pointed out is not only related to the fact that the most frequent type of words in Chinese is bigram, but also is connected with the size of the TREC corpus. To effectively organize the work for document relevancy judgment, methods such as MTC (Carterette *et al.*, 2006) need to be employed.

References

http://trec.nist.gov/

http://trec.nist.gov/pubs/trec10/appendices/measures.pdf

Allan, J., B. Carterette, J. Aslam, V. Pavlu, B. Dachev, and E. Kanoulas. 2007. Million Query Track 2007 Overview. *TREC Notebook*2007.

Bannard, C., T. Baldwin and A. Lascarides. 2003. A Statistical Approach to the Semantics of Verb-Particles. *Proceedings of the ACL-SIGLEX Workshop on Multiword Expressions*, pp.65-72.

Braschler, M., and B. Ripplinger. 2004. How effective is stemming and compounding for german text retrieval? *Information Retrieval*, 7(3/4), 291-316.

Carterette, B., J. Allan and R. Sitaraman. 2006. Minimal Test Collections for Retrieval Evaluation. *SIGIR*2006.

Carterette, B. 2007. Robust Test Collections for Retrieval Evaluation. *SIGIR*2007.

Chang, P., M. Galley and C.D. Manning. 2008. Optimizing Chinese Word Segmentation for Machine Translation Performance. *Proceedings of the Third Workshop on Machine Translation*.

Church, K., P. Hanks. 1990. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics,*, Volume 16, Number 1, March 1990.

Cormack, G., C. Palmer, and C. Clarke. 1998. Efficient Construction of Large Test Collections. *SIGIR 98*.

Feng, H., K. Chen, X. Deng and W. Zheng. 2004. Access Variety Criteria for Chinese Word Extraction. *Computational Linguistics*, 30(1).

Fowler, M. 2003. *UML Distilled: A Brief Guide to the Standard Object Modeling Language.* Addison Wesley.

Foo, S. and H. Li. 2004. Chinese word segmentation and its effect on information retrieval. *Information Processing and Management: an International Journal*, 40(1).

Gao, J., M. Li, A. Wu, and C. Huang. 2005. Chinese word segmentation and named entity recognition: a pragmatic approach. *Association for Computational Linguistics.*

Guenthner, F. and X. Blanco. 2004. Multi-lexemic expressions: an overview. *Linguisticae Investigationes Supplementa.*

Halpern, J. 2000. Is English Segmentation Trivial? *Technical report, CJK Dictionary Institute*.

Hatcher, E. and O. Gospodnetic 2004. *Lucene in Action*. Manning Publications Co.

Huang, X., S. Robertson, N. Cercone, and A. An. 2003. Probability-Based Chinese Text Processing and Retrieval. *Computational Intelligence*, 16.

Kendall, M. 1955. *Rank Correlation Methods*. Hafner.

Kim, S.N. and T. Baldwin. 2007. Detecting Compositionality of English Verb-Particle Constructions using Semantic Similarity. *Proc of the 10th Conference of the Pacific Association for Computational Linguistics*.

Lin, D. 1998. Automatic Identification of Non-compositional Phrases. *In Proceedings of the 37th Annual Meeting of the ACL,* 317-24, College Park, USA.

Jones, Karen Sparck 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* Volume 28, Number 1, 1972, pp.11-21

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

McCarthy, D., B. Keller and J. Carroll. 2003. Detecting a Continuum of Compositionality in Phrasal Verbs. *Proc. Of the ACL-SIGLEDX Workshop on Multiword Expressions*.

Nie, J.Y., J. Gao, J. Zhang and M. Zhou. 2000. On the use of words and N-grams for Chinese information retrieval. *Fifth International Workshop on Information Retrieval with Asian Languages*.Hong Kong.

Packard, J.L. 2000. *Morphology of Chinese: A Linguistic and Cagnitive Approach*. Cambridge University Press.

Peng, F., X. Huang, D. Schuurmans, and N. Cercone. 2002. Investigating the Relationship between Word Segmentation Performance and Retrieval Performance in Chinese IR. *Retrieval Performance in Chinese IR, Coling2002*.

Qu, X., C. Ringlstetter, and R. Goebel. 2008. Targeting Chinese Nominal Compounds in Corpora. *Proceedings of the Sixth International Language Resources and Evaluation*.

Robertson, S., S. Walker, S. Jones, M. Beaulieu, and M. Gatford. 1994. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC 1994)*.

Sag, I.A., T. Baldwin, F. Bond, A. Copestake and D. Flickinger. 2002. Mutliword Expression: A Pain in the Neck for NLP. *Proc. of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics*.

Shi, L., and J. Nie. 2009. Integrating phrase inseparability in phrase-based model. *SIGIR*2009.

Sproat, R., Chilin Shih, William Gale and Nancy Chang 1996. A Stochastic Finite-State Word-Segmentation Algorithm for Chinese. *Computational Linguistics*, 22(3), 1996.

Tao, T., and C. Zhai. 2007. An exploration of proximity measures in information retrieval. *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*.

Xia, F., M. Palmer, N. Xue, M.E. Okurowski, J. Kovarik, F.D. Chiou, S. Huang, T. Kroch, and M. Marcus. 2000. Developing Guidelines and Ensuring Consistency for Chinese Text Annotation. *Proc. of the 2nd International Conference on Language Resources and Evaluation*, Athens, Greece.

Xu, R., Q. Lu, H. Hom, and H.K. Kowloon. 2006. A Multi-stage Chinese Collocation Extraction System. *LECTURE NOTES IN COMPUTER SCIENCE*. Springer.

Xu, Y., C. Ringlstetter, and R. Goebel. 2009. A Continuum-based Approach for Tightness Analysis of Chinese Semantic Units. *Proc. of the 23rd Pacific Asia Conference on Language, Information and Computation*.

Appendix A

Manual Tightness Rank of 300 4grams

The following tables (A.1, A.2, and A.3) show the manual tightness ranks of 300 4-grams for evaluation of our tightness continuum. 4-grams in Rank 1 are transliteration nouns and fixed expressions. 4-grams in Rank 2 are proper nouns and terminologies. 4-grams in Rank 3 are phrases which contain a head noun and a modifier noun. It is assumed that 4-grams in Rank1 are the tightest and 4-grams in Rank3 are the loosest. No relation is set between 4-grams in the same rank. The author is the only annotator.

Table A.1: Manual tightness rank of 300 4-grams: rank 1

Chinese	Pinyin	English		Chinese	Pinyin	English
爱因斯坦	Ai Yin Si Tan	Einstein		花样年华	Hua Yang Nian Hua	colorful time
乌鲁木齐	Wu Lu Mu Qi	Urumqi		澳大利亚	Ao Da Li Yang	Australia
奥林匹克	Ao Lin Pi Ke	Olimpic		摩托罗拉	Mo Tuo Luo La	Motorala
阿里巴巴	A Li Ba Ba	Alibaba(A company's name)		马来西亚	Ma Lai Xi Ya	Malaysia
孟加拉国	Men Jia La Guo	Bangladesh Country		巴基斯坦	Ba Ji Si Tan	Pakistan
花花公子	Hua Hua Gong Zi	Play Boy		可口可乐	Ke Kou Ke Le	CocaCola
哈利波特	Ha Li Bo Te	Herry Potter		宇多田光	Yu Duo Tian Guang	The same as Pinyin(person's name)
呼和浩特	Hu He Hao Te	HohHot		丝绸之路	Si Chou Zhi Lu	The Silk Road
风情万种	Feng Qin Wang Zhong	exceedingly fascinating and charming		欢天喜地	Huan Tian Xi Di	wild with joy
倾家荡产	Qing Jia Dang Chan	be reduced to poverty and ruin		没完没了	Mei Wan Mei Liao	be endless
一枝独秀	Yi Zhi Du Xiu	outshine others (Literal: One branch of the tree is particularly thriving)		改朝换代	Gai Chao Huan Dai	Things change (Literal: dynastic changes)
白手起家	Bai Shou Qi Jia	start from scratch		不堪回首	Bu Kan Hui Shou	find it unbearable to recall

Table A.2: Manual tightness rank of 300 4-grams: rank 2 (partial)

Chinese	Pinyin	English		Chinese	Pinyin	English
清华大学	Qing Hua Da Xue	Qinghua University		武汉大学	Wu Han Da Xue	Wu Han University
浙江大学	Zhe Jiang Da Xue	Zhejiang University		湖南大学	Hu Nan Da Xue	Hu Nan University
交通大学	Jiao Tong Da Xue	Jiao Tong University		科技大学	Ke Ji Da Xue	Keji University
北京大学	Bei Jing Da Xue	Beijing University		渣打银行	Zha Da Yin Hang	The Chartered Bank
商业银行	Shang Ye Yin Hang	The Commercial Bank		人民银行	Ren Min Yin Hang	The People's Bank
工商银行	Gong Shang Yin Hang	Industrial and Commercial Bank		人民日报	Ren Min Ri Bao	People's Daily
中华民国	Zhong Hua Min Guo	Republic of China		本草纲目	Ben Cao Gang Mu	An Outline Treatise of Medical Herbs
黑龙江省	Hei Long Jiang Shen	Heilongjiang Province		计划生育	Ji Hua Sheng Yu	Planned Parenthood
高尔夫球	Gao Er Fu Qiu	Golf Ball		观音菩萨	Guan Yin Pu Sa	the Goddess of Mercy
经济学家	Jing Ji Xue Jia	economist		康熙大帝	Kang Xi Da Di	Kangxi emperor

Table A.3: Manual tightness rank of 300 4-grams: rank 3 (partial)

Chinese	Pinyin	English		Chinese	Pinyin	English
美国大学	Mei Guo Da Xue	US's University		外资银行	Wai Zi Yin Hang	foreign bank
长安汽车	Chang An Qi Che	Chang'an Vehicle		交易市场	Jiao Yi Shi Chang	trade market
金属材料	Jin Shu Cai Liao	metal material		义务教育	Yi Wu Jiao Yu	compulsory education
人寿保险	Ren Shou Bao Xian	life insurance		医疗设备	Yi Liao She Bei	armarium
制造公司	Zhi Zhao Gong Si	manufacturing company		集团公司	Ji Tuan Gong Si	group company
食品公司	Shi Ping Gong Si	food company		外贸公司	Wai Mao Gong Si	foreign trade corporation
中医学院	Zhong Yi Xue Yuan	college of traditional Chinese medicine		技术学院	Ji Shu Xue Yuan	tech college
风险管理	Feng Xian Guan Li	risk management		投资管理	Tou Zi Guan Li	management of investment
工商管理	Gong Shang Guan Li	business administration		中国女排	Zhong Guo Nv Pai	Chinese Women's Volleyball Team
中国汽车	Zhong Guo Qi Che	Chinese Vehicle		背景音乐	Bei Jing Yin Yue	background music

Appendix B

Test Collection System

Test collections are important for information retrieval evaluation. There are three components that define an IR test collection: a query set, a corpus from which relevant documents are retrieved, and relevance judgements for the documents in connection with the queries. As the current test collections are not suitable to deeply analyze the effect of Chinese segmentation upon IR, we created our own test collection. Please refer to Section 4.2 for the description of our query set and corpus. In the following, I will describe the system and its progress to get relevance judgements for documents.

Figure B.1 shows the user interface of the test collection system. The input of the system is the top n results of several IR systems. First, we choose a query by click the Button "Open." (Figure B.2 and Figure B.3), choose the directory where the documents to be judged are and the judgement results will be. Usually the documents to be judged are combination of top 100 results of several IR systems. As Figure B.4 shows, the query is "中原油田" (Midland oil field). Then we choose "Collection Method" as "MinTestCollection" (Carterette *et al.*, 2006). After we click the Button "Load", it will load in documents which need to be judged next. Query characters in documents are in red front to help annotators speed up. If the document is not relevant, we click the Button "Irrel...", the label of the document will turn red (Figure B.5). After judging a document, we click the Button "Next" to load in another document (Figure B.6). If it is relevant, we click the Button "Rele...", the label of the document will turn blue (Figure B.7). Judge other documents until the Button "Next" is not available, i.e. you have judged 100 documents or all the documents retrieved by those IR systems.

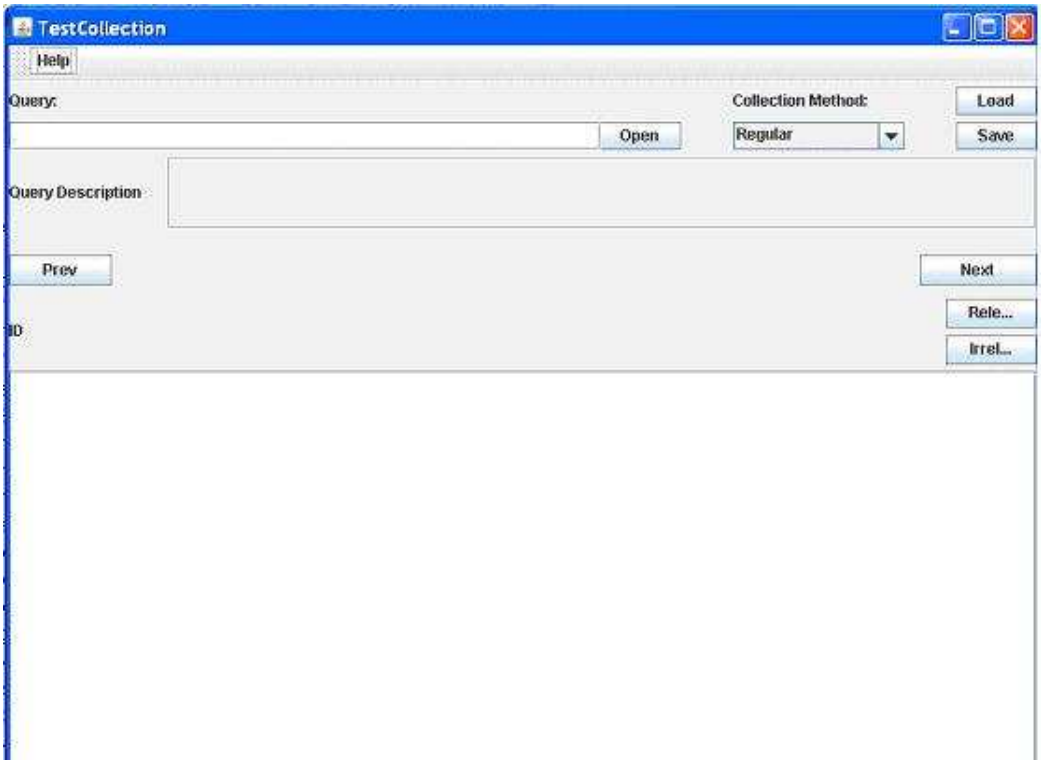

Figure B.1: Test collection system user interface (1).

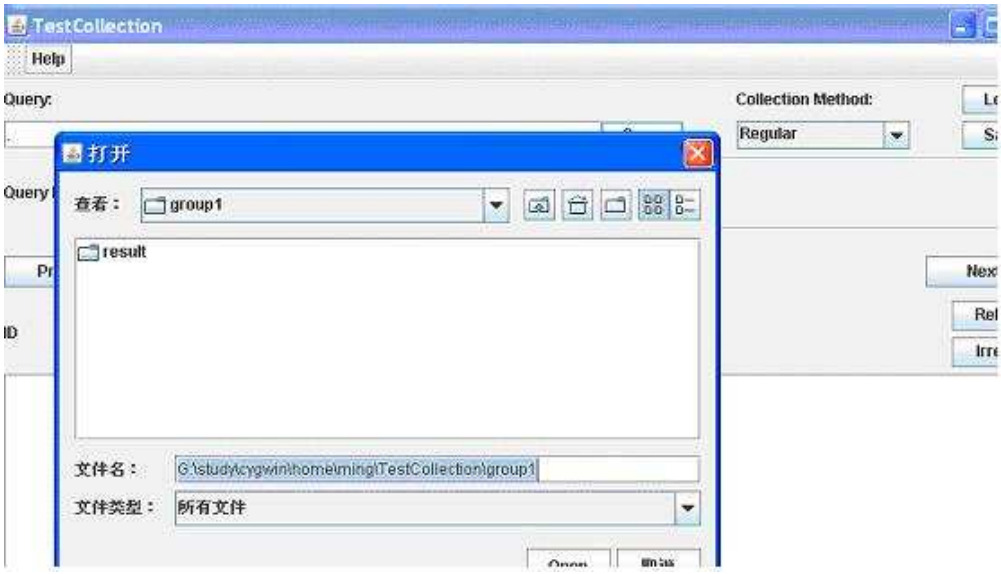

Figure B.2: Test collection system user interface (2).

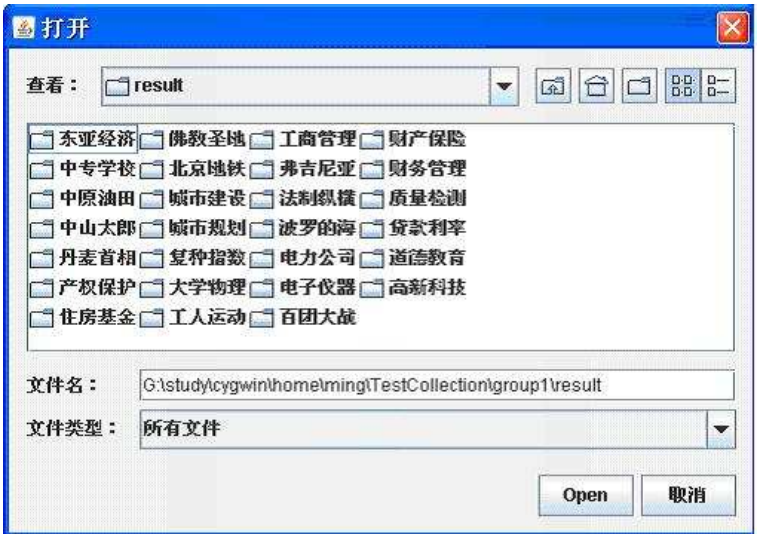

Figure B.3: Test collection system user interface (3).

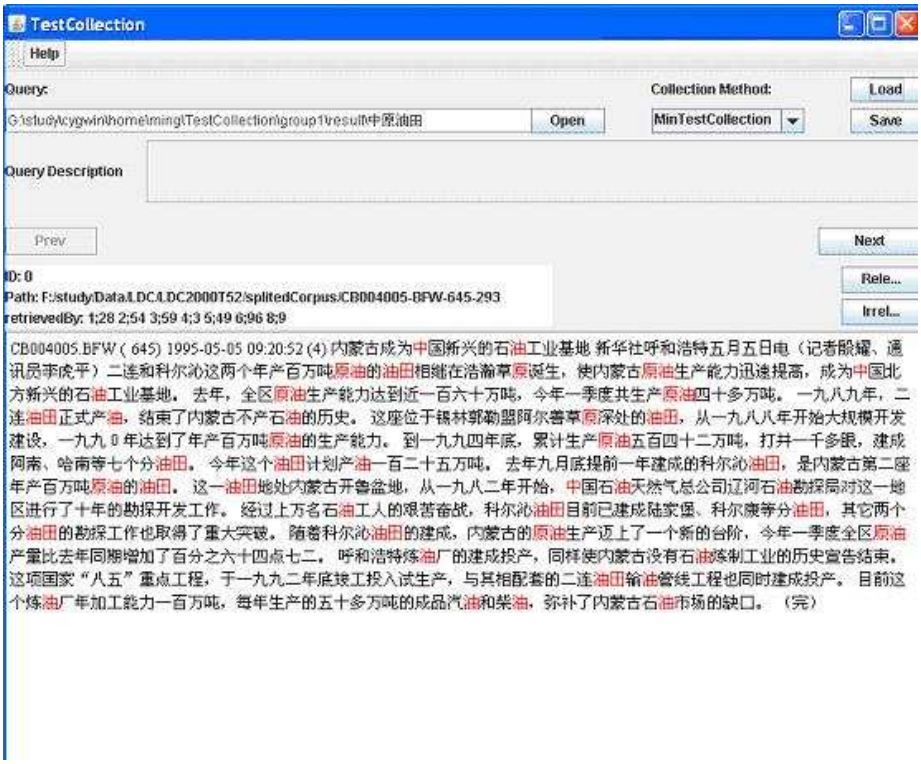

Figure B.4: Test collection system user interface (4).

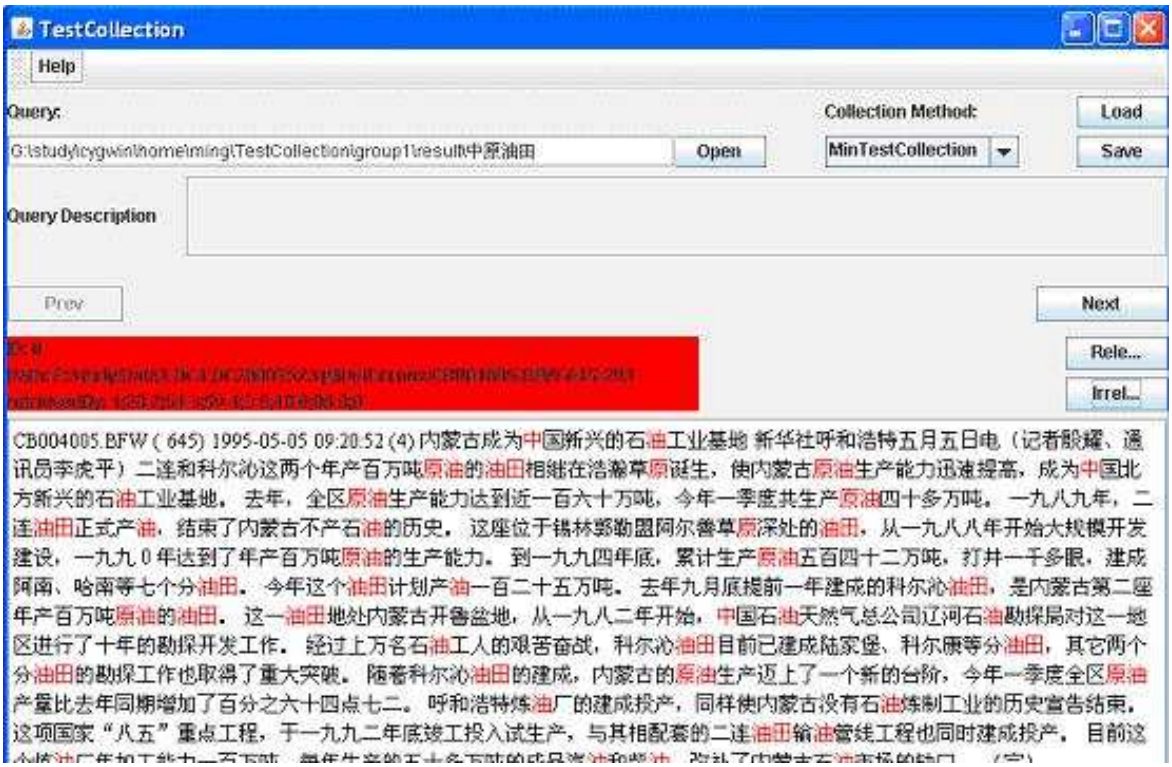

Figure B.5: Test collection system user interface (5).

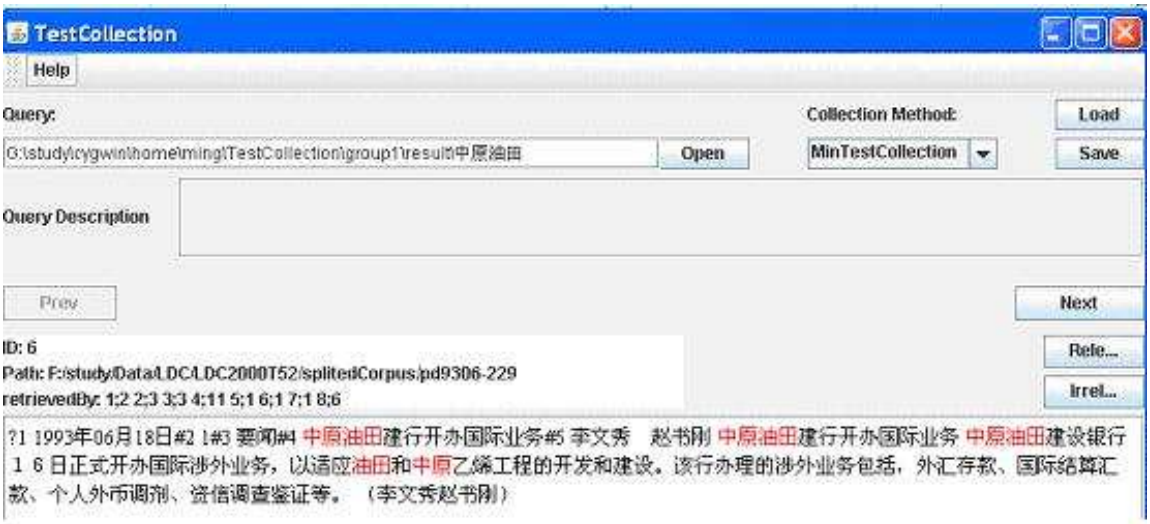

Figure B.6: Test collection system user interface (6).

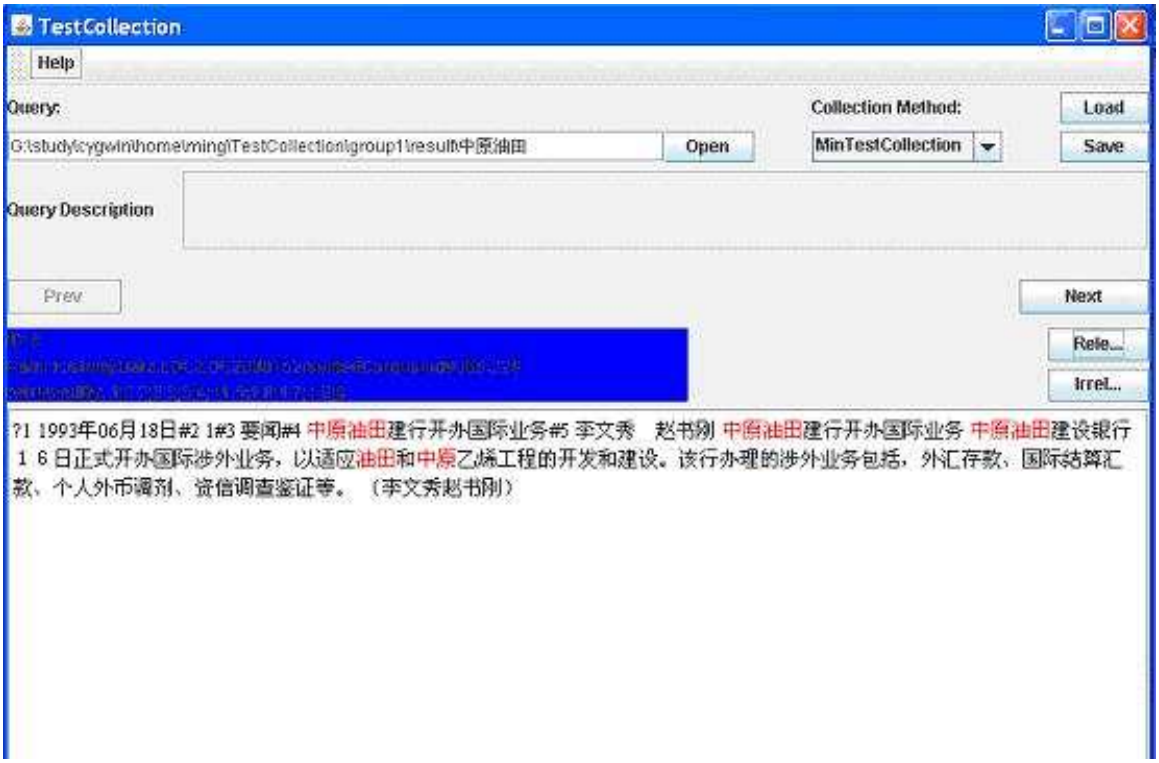

Figure B.7: Test collection system user interface (7).