

2011.2785.1

Université de Montréal

Parallel Text Mining for Cross-Language Information Retrieval  
Using a Statistical Translation Model

par

Jiang Chen

Département d'informatique et de recherche opérationnelle  
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures  
en vue de l'obtention du grade de  
Maître en Informatique

Avril, 2000

©Jiang Chen, 2000



QA

76

U54

2000

N. 031



Université de Montréal  
Faculté des études supérieures

Ce mémoire intitulé:

Parallel Text Mining and Cross-Language Information Retrieval  
with a Statistical Translation Model

présenté par:

Jiang Chen

a été évalué par un jury composé des personnes suivantes:

Jean-Yves Potvin (président-rapporteur)

Jian-Yun Nie (directeur de recherche)

Guy Lapalme (membre du jury)

Mémoire accepté le 29-05-2000

To Janie

# Acknowledgement

This work has been a challenging experience in which I learned many exiting technologies and excellent works of others. Throughout the year I received a lot of support from the people in the laboratory RALI in various ways. I'm glad I worked for my master's in such an enviroment.

First of all, I would like to thank Prof. Jian-Yun Nie, my supervisor, who conceived this work and guided me all the way through. I also owe my gratitude to Michel Simard and Pierre Plamondon for their good work of parallel text alignment method/program and the implementation of the statistical translation model, which are the two most fundamental tools I used in this work. I would especially thank Michel and Philippe Langlais for all the explanations and discussions that have proved to be very beneficial to me. My appreciation also goes to Prof. Guy Lapalme and Elliott Macklovitch, director of RALI, for all their encouragement.

# Abstract

Statistical translation model is one of the possible tools to translate queries from a language to another for Cross-Language Information Retrieval (CLIR). In comparison with a machine translation (MT) system, this approach is much easier to implement. Compared with a bilingual dictionary, it can also produce translations of higher quality.

However, a necessary condition for the building of a translation model is the availability of a large corpus of parallel texts from which the model can be trained. Such a corpus is not always available for many language pairs such as English and Chinese. We observe that the Web contains many parallel pages for many language pairs. They can be used to train translation models if there are means to identify them. The system we constructed - PTMiner - aims to identify automatically parallel texts from the Web. This system is relatively language-independent so that it can be easily adapted to different language pairs. We successfully used the system to gather a large set of parallel texts in English and Chinese. Two translation models (in two directions) are trained from them. The models are then used to translate CLIR queries between Chinese and English. Our experiments show that the translation models can produce word translations of reasonable quality. When they are applied to CLIR tasks, we can obtain similar performances to the case using commercial MT systems. This work shows the feasibility of collecting large parallel corpora from the Web for CLIR purposes. It paves the way for a low cost yet high quality means to CLIR. Some papers [CN00b, CN00a, CN00c] about this work have been accepted by various conferences.

In this thesis, we first describe the principle and the implementation of PTMiner. The parallel texts gathered from the Web will be used to train translation models. These models will finally be used in CLIR tasks.

# Résumé

Une façon possible de traduire des requêtes dans la Recherche d'Information Translinguistique (RIT) est d'utiliser un modèle de traduction statistique. Par rapport à un système de traduction automatique (TA), cette approche est beaucoup plus facile à réaliser. En comparant avec un dictionnaire bilingue, elle peut aussi produire des traductions de meilleure qualité.

Cependant, un prérequis pour la construction d'un tel modèle est la disponibilité d'un grand corpus de textes parallèles à partir duquel le modèle est entraîné. Pour beaucoup de paires de langues telles que l'anglais et le chinois, un tel corpus est difficile à obtenir. Nous observons que le Web contient beaucoup de pages parallèles. Ces pages peuvent servir pour l'entraînement de modèles de traduction si elles peuvent être identifiées automatiquement. C'est dans ce but que nous avons construit le système PTMiner pour collecter automatiquement des pages parallèles du Web. Ce système est relativement indépendant des langues, et peut être ainsi facilement adapté pour différentes paires de langues. Nous avons réussi à obtenir un grand corpus de textes parallèles en chinois et en anglais en utilisant ce système. Deux modèles de traduction (dans deux directions) ont été entraînés. Ces modèles ont été utilisés pour traduire des requêtes entre le chinois et l'anglais. Nos résultats expérimentaux montrent que les traductions produites par ces modèles sont d'une qualité raisonnable. Quand ils sont utilisés pour la RIT, nous pouvons obtenir des performances similaires à l'approche qui utilise des systèmes de TA commerciaux. Ce travail prouve qu'il est possible de collecter de grands corpus parallèles du Web pour les besoins de la RIT. Nous pouvons envisager la réalisation d'un moyen de traduction de requêtes bon marché mais de



bonne qualité. Dans ce mémoire, nous décrivons d'abord le principe et l'implantation de PTMiner. Les textes parallèles obtenus sont ensuite utilisés pour entraîner des modèles de traduction. Finalement, ces modèles sont utilisés pour la RIT.

Certains articles [CN00b, CN00a, CN00c] concernant ce travail ont été acceptés par plusieurs conférences.

Recherche d'information translinguistique

Modèle de traduction statistique

# Contents

Acknowledgements	i
Abstract	ii
Résumé	iv
Contents	viii
List of Figures	x
List of Tables	xi
<b>1 Introduction</b>	<b>1</b>
<b>2 Parallel Text Mining</b>	<b>6</b>
2.1 Introduction to Knowledge Discovering Technologies . . . . .	7
2.1.1 Technology Overview . . . . .	7
2.1.2 Intelligent Miner for Text – IBM’s Text Mining Solution . . . .	11
2.1.3 STRAND . . . . .	12
2.2 Parallel Text Mining Algorithm . . . . .	13
2.2.1 About the Search Engines . . . . .	14
2.2.2 Candidate Sites Search . . . . .	15
2.2.3 File Name Fetching . . . . .	16
2.2.4 Host Crawling . . . . .	17
2.2.5 Pair Scan (by Naming Patterns) . . . . .	19

2.2.6	Filtering . . . . .	21
2.3	PTMiner – A Multi-Tier Distributed Text Miner . . . . .	23
2.3.1	Adopted Technologies . . . . .	24
2.3.2	PTMiner Architecture and Implementation . . . . .	26
2.4	Mining Results Analysis . . . . .	35
2.5	Summary . . . . .	39
<b>3</b>	<b>Training the English-Chinese Statistical Translation Model</b>	<b>40</b>
3.1	Related Chinese Information Processing Techniques . . . . .	42
3.1.1	Coded Chinese Character Set Standards . . . . .	42
3.1.2	Related techniques . . . . .	45
3.2	Aligning the Parallel Corpus . . . . .	47
3.2.1	Introduction to Text Alignment Methods . . . . .	48
3.2.2	Using Markups as Cognates . . . . .	51
3.3	Chinese Segmentation and English Expression Extraction . . . . .	57
3.3.1	Chinese Segmentation . . . . .	57
3.3.2	English Expression Extraction . . . . .	58
3.4	Translation Model Training . . . . .	59
3.4.1	Principles of the Statistical Model . . . . .	60
3.4.2	Analysis on Evaluation Lexicons . . . . .	61
3.5	Summary . . . . .	70
<b>4</b>	<b>CLIR Experiments</b>	<b>71</b>
4.1	CLIR Approaches . . . . .	71
4.2	Chinese-English CLIR . . . . .	72
4.2.1	The Collection . . . . .	73
4.2.2	Mono-Lingual IR Results . . . . .	73
4.2.3	CLIR Using Translation Model . . . . .	73
4.2.4	CLIR Using Dictionary . . . . .	76
4.2.5	Combining Translation Model and Dictionary . . . . .	76

4.3	English-Chinese CLIR . . . . .	78
4.3.1	The Collection . . . . .	79
4.3.2	Mono-Lingual IR Results . . . . .	79
4.3.3	CLIR Using Translation Model . . . . .	79
4.3.4	CLIR Using Dictionary . . . . .	81
4.3.5	Combining Translation Model and Dictionary . . . . .	81
4.3.6	CLIR Using MT System . . . . .	82
4.4	Comparison to English-French CLIR . . . . .	83
4.5	Summary . . . . .	85
<b>5</b>	<b>Conclusions</b>	<b>87</b>
	<b>Bibliography</b>	<b>90</b>

# List of Figures

2.1	The workflow of the mining process. . . . .	14
2.2	A pair of parallel pages linking to each other. . . . .	15
2.3	Candidate sites search using Web search engines. . . . .	17
2.4	Parallel pairs in a directory tree. . . . .	20
2.5	The architecture of PTMiner. . . . .	27
2.6	The ERD of the PTMiner database. . . . .	28
2.7	PTMonitor showing messages from the servers. . . . .	31
2.8	PTMonitor showing the crawler table of the database. . . . .	32
2.9	PTMonitor showing the result set of the user defined query. . . . .	32
2.10	PTMonitor showing mining progress. . . . .	33
2.11	A network view of the PTMiner system. . . . .	34
2.12	The distribution of true pairs on different length differences. . . . .	37
2.13	Precision obtained applying different criteria on length difference. . . . .	38
3.1	Pre-training procedures. . . . .	41
3.2	Sentences are assembled as the training source. . . . .	41
3.3	Convert biaodian (Chinese punctuations) to its corresponding ASCII punctuation. . . . .	46
3.4	An alignment example using pure length-based method. . . . .	53
3.5	An alignment example using pure length-based method (continued). . . . .	54
3.6	An alignment example considering cognates. . . . .	55
3.7	An alignment example considering cognates (continued). . . . .	56
3.8	E-C translations. . . . .	64

3.9	C-E translations. . . . .	65
3.10	Effect of English stop list in C-E translation. . . . .	67
3.11	Effect of Chinese stop list in C-E translation. . . . .	67
3.12	More E-C translations. . . . .	68
4.1	English-French CLIR results. . . . .	84

# List of Tables

3.1	Statistics of the training source. . . . .	61
3.2	Precision of testing models. . . . .	63
4.1	Results of CLIR using translation model (translate by queries). . . .	75
4.2	Results of CLIR using translation model (translate by words). . . . .	76
4.3	Results of CLIR using a dictionary. . . . .	77
4.4	CLIR Results of combining queries translated by translation model (without weight) and dictionary. . . . .	77
4.5	CLIR Results of combining queries translated by translation model (keeping weight) and dictionary. . . . .	78
4.6	Results of CLIR using translation model. . . . .	79
4.7	Results of CLIR using an online English-Chinese dictionary. . . . .	81
4.8	CLIR Results of combining queries translated by translation model (without weight) and dictionary. . . . .	81
4.9	CLIR Results of combining queries translated by translation model (keeping weight) and dictionary. . . . .	82
4.10	Best combining ratios for E-C and C-E CLIR. . . . .	82

# Chapter 1

## Introduction

In this thesis, we describe our work of building a Chinese-English statistical translation model trained by a parallel corpus collected from the Web, and using this model in Cross-Language Information Retrieval (CLIR).

Traditionally, information retrieval (IR) systems have been widely used by libraries, scientific organizations, and corporations to provide access to books, journals, and other documents. The emergence and fast growth of the Web in the last decade provided great demand and new challenges to IR systems.

As defined by Salton and McGill, “Information retrieval is concerned with the representation, storage, organization, and accessing of information items. Items found in retrieval systems are characterized by an emphasis on narrative information. Such narrative information must be analyzed to determine the information content and to assess the role each item may play in satisfying the information needs of the system users.” [SM83] With the development of the Internet, various search engines have been put into service. Although these search engines look different from traditional IR systems, their basic functionalities are the same, except that the document collection is dynamic and changing. Another difference in the Internet environment is the use of different languages in the same collection. Search engines are faced with the problem of Cross-Language Information Retrieval (CLIR). In a broad sense, CLIR refers to retrieving relevant documents in any language from a given query. In a narrower



sense, it means retrieving documents in one particular language which is different from the query language. The problem involved in the narrow CLIR is fundamental in any CLIR. We believe that before building a CLIR system in the broad sense, the narrow CLIR task should be implemented correctly. This thesis is concerned with CLIR in the narrow sense.

In order to match a document and a query in two different languages, either the document or the query should be translated. We then have two approaches to CLIR using either document translation [DLL96] or query translation [Kwo99, NSID99]. Intuitively, it is more flexible and easier to implement query translation than document translation. Although some believe that document translation may be more accurate because of more contextual information, the previous experiments did not show that it always performs better than query translation. In this thesis, we focus on the query translation approach.

There are basically three groups of query translation approaches: using a machine translation (MT) system, using a bilingual dictionary, and using a statistical translation model.

MT systems have been the object of research and development of over 50 years. Although there are several commercial systems for a number of major language pairs, many language are not covered (correctly) by them. There does not seem to be a major break-through in this field to make the construction of MT systems much easier in the near future. Therefore, MT systems are possible but costly means for query translation in CLIR. In addition, as stated in [NSID99], the current MT approaches are not completely compatible to CLIR requirements:

- They spend much effort to generate syntactically correct sentences. This effort is irrelevant to the current practice in IR which is mainly based on keywords.
- They choose only one translation word in the target sentence even though there are several synonyms. In IR, one is interested in adding synonyms (and related words) into the query so that more relevant documents may be retrieved.

The dictionary-based approach is much simpler: it only looks into the dictionary

to determine all the translations. The problem with it is word ambiguity because all the meanings of the source word are mixed up in the translations. In practice, dictionary-based approaches achieve much poorer performances than MT approaches.

The third approach uses a large set of parallel texts to estimate a statistical translation model. The principle is: the more two words co-occur in parallel sentences (one being the translation of the other), the more probable they are translation of each other. The experiments of Nie et al. [NSID99] for English-French CLIR showed that the approach can achieve a high performance comparable to the MT approach. In this thesis, we extend the approach to English-Chinese CLIR.

A necessary condition of the approach is the availability of a large corpus of parallel texts. For some language pairs (e.g. French and English) such corpora exist; but for many others, they do not. We notice that the WWW contains many parallel pages, in most cases between English and another language. If we can gather these pages automatically, we then can construct parallel corpora at a low cost. Our first goal in this work is to develop a mining system for parallel texts from the Web. The system is called PTMiner (for Parallel Text Miner). It is relatively language independent so that it can be easily adapted to different language pairs (always with English). Our work focuses on the mining of Chinese-English parallel pages.

Another goal of our work is to test the effectiveness of a translation model trained with the parallel pages. Therefore, we also carry out the following two tasks:

- Establish a Chinese-English translation model trained by the Web corpus and research on specific issues in the training process, such as parallel text alignment;
- Evaluate the translation model's performance in CLIR and compare (combine) with other CLIR approaches.

Our experiments show that when the translation model is used in combination with a small bilingual dictionary, we can achieve a performance comparable to that using a MT system. This result shows that for CLIR, we can build a means for query translation at a much lower cost than an MT system.

The major contribution of this work is that it is a complete implementation of the idea: cross-language information retrieval using a statistical translation model trained by Web-collected parallel corpus. It not only experimented a low cost yet effective CLIR approach, but also provided some by-product such as the parallel text mining method/implementation, and experiences of applying many existing natural language processing techniques. In the translation model training and CLIR experiments many existing programs/system such as the Chinese segmentation program, the English expression extraction program, the translation model pretraining scripts, the parallel text alignment program, the implementation of the statistical model and the Smart IR system, are adopted.

In the next three chapters, we will present the details of the three phases of this work, namely, parallel text mining, statistical translation model training, and Chinese-English CLIR experiments.

In Chapter 2, we will first give an introduction on some knowledge discovering technologies. Then we will explain our parallel text mining algorithm, which consists of candidate sites search, file name fetching, host crawling, pair scan and filtering. The PTMiner system implements this algorithm in a distributed model. Its architecture and components will be introduced. We will also give an analysis on the mining results.

Chapter 3 will cover the training process of the Chinese-English translation model. Many language specific techniques such as Chinese character set standards, Chinese word segmentation, and English expression extraction, are involved in the pre-training process. Among all the steps, parallel text alignment is a particularly interesting and challenging one. We adopted the alignment method of Simard et al. [SFI92] which considers both sentence length and cognates (HTML markups in this case). To evaluate the precision of the trained models, we examined the evaluation lexicons.

In Chapter 4, we will not only present the results of both E-C and C-E CLIR using the translation models trained by the Web corpus, but also compare our approach with dictionary-based approach and MT systems. We will also try to combine the translations given by a statistical model and some dictionary so as to improve CLIR

performance.

Chapter 5 will be a summary and conclusion of this work.

Some papers [CN00b, CN00a, CN00c] about this work have been accepted by various conferences.

# Chapter 2

## Parallel Text Mining

Data mining, text mining and other knowledge discovering techniques have become an attractive research area in the past years. The explosive growth of information offered potential solutions to more and more problems. Parallel text mining, which aims to discover pairs of text that are translations for each other, is one of the many cases that people discover various kinds of knowledge from available information to solve specific problems.

In this chapter, we will first introduce the main technologies proposed to discover knowledge from electronic information available in different forms. As a specific case, the package of text mining solutions of IBM, “Intelligent Miner for Text”, will be discussed. For the purpose of producing a Chinese-English parallel text corpus, a system named PTMiner (Parallel Text Miner) was developed. We will explain the underlying mining algorithm of this system as well as its implementation using distributed computing and database technology.

## 2.1 Introduction to Knowledge Discovering Technologies

### 2.1.1 Technology Overview

As the volume and importance of electronic information sources such as corporate databases, data warehouses, intranet documents, business emails and of course the World Wide Web, are growing explosively, knowledge discovering has become a highly demanding task. In the recent years many researchers and companies have focused on this task and a good progress has been made.

According to the way they are organized, electronic information sources can be categorized into:

- Structured data such as databases;
- Semi-structured texts such as hyper texts and emails;
- Unstructured text such as articles and reports.

Various technologies are designed to deal with the information with different natures. We will describe some of them in the following.

#### **OLAP and Data Mining**

Two technologies, OLAP (OnLine Analytical Processing) and data mining [HCC<sup>+</sup>97], are applied by companies to explore the strategic value of the huge amount of highly structured data in their databases and data warehouses.

OLAP is a set of techniques for analyzing data in data warehouses. A data warehouse is a database designed to support decision making in an organization. It stores a large collection of subject-oriented, integrated, historical and static data. OLAP servers organize data into multidimensional hierarchies to realize high-speed data analysis.

Data mining is defined as the process of discovering relationships and patterns by scanning databases or other information repositories. Han [Han99] describes the major tasks of data mining as:

- Class description of data collection;
- Discovery of association relationships;
- Data classification;
- Prediction of missing data or value distribution of attributes in a set of objects;
- Clustering similar objects in a collection;
- Time-series analysis.

Both based on underlying database management systems, OLAP and data mining serve different yet complementary functions. OLAP provides a top-down view of the data while data mining is a bottom-up discovery. They are often combined together for better performance [Han97].

### **Text Mining**

When it comes to semi-structured or unstructured data, the term “text mining” is frequently used. Research [For95] showed that unstructured data is becoming the predominant data type available online. Companies and organizations not only have large databases, but also large and increasing amount of online documents such as:

- Intranet documents (announcements, meeting memos, working emails, etc.);
- Customer emails containing business information;
- Technical reports.

These online documents carry a great deal of information that could provide crucial help to strategic decision making. Needless to mention, the Web is another incredibly large and heterogeneous repository of online documents. Making effective use of the

potential value of these information resources has become a very interesting subject. Text mining may concern the following tasks:

- Information extraction;
- Document or collection summarizing;
- Document categorization;
- Clustering of collections;
- Text searching.

The problem with text mining is that unlike tabular records in databases, documents are not structured and normalized so that they could be easily recognized by computers. The lack of structure raises the difficulty of uncovering the implicit knowledge inside the documents. It is hard to extract and represent abstract concepts from a natural text because the same concept may be expressed in many ways.

## **Web Mining**

The World Wide Web, with its unique features such as semi-structured hypertext format, links between web pages and a lot more, distinguishes itself from other on-line document resources. This fact makes the term “web mining” and the techniques specifically tailored for mining information from the web necessary. Cooley et al. [CMS97] has given a taxonomy of web mining. They categorize web mining into web content mining and web usage mining. Web content mining refers to automatic searching and discovering information from the Web. Web usage mining is the discovery of user access pattern from Web servers. Web content mining is on the client side while web usage mining is on the server side.

Web content mining extended the functions of traditional web search engines. Web content miners not only do simple search for relevant documents but also try to provide structural and implicit information by categorizing, filtering and interpreting Web documents. To achieve these functions, people either develop intelligent web



agents [BDH<sup>+</sup>94, BSY95, DEW96, PE95] for various demands or set up multilevel databases based on the Web information and Web query systems [KS95, ZH98]. Intelligent Web agents are developed to search for relevant information using domain characteristics and user profiles. PTMiner, an intelligent parallel text miner we developed to search for Chinese-English (or other language pairs) parallel texts from the Web, belongs to this category.

On the other hand, Web usage mining is heavily demanded by the organizations to collect information from the daily transactions of their Web servers. The information includes customer or potential customer information, product marketing information, and the effectiveness of the site itself, among other things. Some sophisticated systems [ZXH98, MJHS96] were developed for the research of pattern discovery and analysis from Web server access logs.

### **Search vs. Discovery**

What is the difference between text mining and traditional information retrieval? What is the difference between Web miners and Web search engines? The answer is that the primary goal of text mining or Web miners is not locating documents, but extracting valuable and relevant textual information and presenting them in a structural organization. Search and discovery are two complementary processes. Discovery is often based on searching results. It exploits the searching results for more valuable information according to specific characteristics of the discovery task.

As an example, the PTMiner system in this project relies on the major Web search engines to obtain potential bilingual sites and URLs of the documents in these candidate sites. From the searching results, the PTMiner then discovers parallel documents according to common naming patterns. We will explain the details of the mining algorithm in the next section.

## 2.1.2 Intelligent Miner for Text – IBM’s Text Mining Solution

Text mining is still in its early era. However, because of its important value for enterprises and organizations, commercial software has already been emerging. IBM’s Intelligent Miner for Text [IMT99, Tka98] and the SemioMap of Semio Corp. [Sem98] are two of the commercial text mining products intended to give enterprises a complete solution of knowledge management.

The features and quality of commercial text mining software reflect the current state of the art of this field. Here we would like to discuss the tools contained in the package of the Intelligent Miner for Text of IBM, which is leading in application and research of text mining technologies. Among others, we are particularly interested in the Web Crawler package, which inspired the design of the architecture of the PTMiner system.

The Intelligent Miner for Text consists of several sub-packages developed for various functions. We describe them in the following paragraphs.<sup>1</sup>

### Text Analysis Tools

This package contains a set of tools that “automatically identify the language of a document, create clusters as logical views, categorize documents, summarize documents, and extract relevant textual information such as proper names and multi-word terms”.

### IBM Text Search Engine

In addition to being an information retrieval system which “does in-depth document analysis during indexing, and offers sophisticated query enhancement and result preparation to make text retrieval significantly more informative and effective”, the text search engine is enhanced with “mining functionality and capabilities to visualize results”. It also provides linguistic analysis supporting tens of languages.

---

<sup>1</sup>The quoted parts of the following introduction are extracted from [IMT99].

## NetQuestion Solution

Combining the functionality of the Text Search Engine and the Web Crawler, it is a text search solution designed for documents located in local and internet Web servers.

## Web Crawler

The Web Crawler package includes a Web Crawler Toolkit that allows users to develop their own Web crawlers according to various needs. A ready-to-run implementation of Web crawler is also included. Together with the NetQuestion Solution package, it gives users ability to have convenient and powerful access to relevant information in internet/intranet Web servers.

From the implementation point of view, the following features of IBM Web Crawler helped the design of the PTMiner system:

- The crawler can run in a single machine or in user-specified number of machines in parallel.
- The meta-data of crawling results such as URL, size and date is stored in DB2 database. It greatly facilitates the maintenance of crawling process and the management of crawling results.
- It provides command-line or GUI interfaces to monitor and control crawling process.

### 2.1.3 STRAND

One of the related works on Web parallel text mining is the STRAND (Structural Translation Recognition for Acquiring Natural Data) approach of Resnik. In his preliminary investigation of mining the Web for parallel text [Res98], Resnik proposed a mining algorithm which consists of candidate generation and candidate evaluation. In the first step, the method simply sends a Boolean query in the form:

*anchor : language1 AND anchor : language2*

to AltaVista to locate pages that potentially point to pairs of parallel pages. Obviously this simple method can only catch a small part of all the parallel pages. There are a lot of parallel pages that do not satisfy this condition. When a large size corpus is demanded as in our case, we have to search more thoroughly.

In candidate evaluation, the STRAND approach aligns each pair of documents using HTML markups and then computes the correlation for the lengths of the aligned chunks (not including the markups). The significance of this correlation is used as the criterion to identify parallel text.

## 2.2 Parallel Text Mining Algorithm

The PTMiner system is an intelligent Web agent that is designed to search for parallel text from the Web. We will explain its mining algorithm in this section and its implementation in the next.

The general idea of the algorithm is to first locate the bilingual Web sites that could possibly contain parallel texts. Then we search inside each candidate site according to general naming patterns of file names. We can summarize the mining process of PTMiner as the following steps:

- 1 Candidate sites search – Search from the Web search engines for the candidate sites that could contain parallel pages;
- 2 File name fetching – For each candidate site, fetch the URLs of Web pages that are indexed by the search engines;
- 3 Host crawling – Starting from the URLs collected in the last step, crawl each candidate site separately for more URLs;
- 4 Pair scan – From the obtained URLs of each site, scan for possible parallel pairs according to common naming patterns;
- 5 Download and verifying – Download the parallel pages, determine file size, language, and character set of each page, and thus filter out non-parallel pairs.

Fig. 2.1 illustrates the workflow of the mining process.

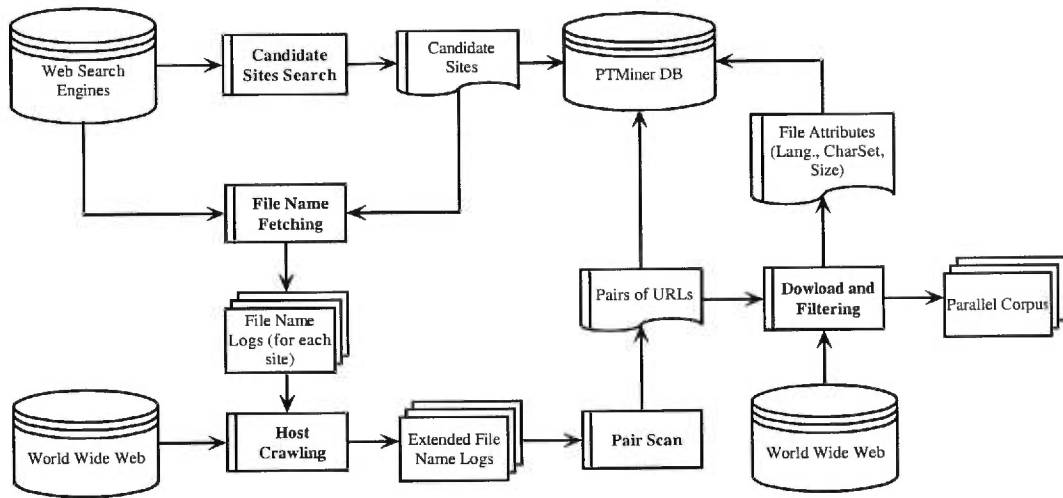


Figure 2.1: The workflow of the mining process.

### 2.2.1 About the Search Engines

Before we go through the details of each step, it is necessary to give a survey on the Web search engines, which play an important role in this algorithm. We use the search engines to take advantage of their huge amount of indexed documents. PTMiner relies on search engines to locate candidate sites and fetch URLs of each site to provide a starting URL set for host crawling. Fetching URLs from search engine is much more efficient than crawling the site itself. Actually, in the first run of PTMiner, we found 5000 pairs even without applying host crawler.

Search engines constantly visit Web sites on the Internet in order to create catalogs of web pages. They usually run some robots to automatically discover new Web pages and index them. Currently there are 10-20 major search engines on the Web with various index sizes and functions. Latest research [Ano99a] shows that current existing engines cover around 16% of all the Web. Among them AltaVista and Northern Light are the two largest ones in terms of pages indexed. They also have some features that are crucial to parallel text mining such as keyword searching

and language identification. A new search engine *alltheweb.com* was launched in early August 1999. It aims to cover every page on the Web. At the starting point, it already covered 25% of the Web, i.e., 200 million URLs. Full coverage was expected by the end of 1999 [Ano99a]. Despite its high coverage, *alltheweb.com* is weak at query functionality at this moment.

Both AltaVista and Northern Light have language identification ability. It makes it possible for us to locate candidate bilingual sites. For Chinese-English parallel text search AltaVista is especially helpful not only because it is the only search engine that can identify Chinese pages but also because it can return results in all the character sets of Chinese. AltaVista does this by translating the pages it finds into Unicode, which can store characters for all languages.

### 2.2.2 Candidate Sites Search

In the sites where parallel text exists, there are normally some pages in one language containing links to its version in the other language. Those links' anchor texts<sup>2</sup> usually indicate that. For example, (Fig. 2.2) in some English page there may be a link to its Chinese version with the anchor text "Chinese Version", "in Chinese", and so on. The same phenomenon can be observed in Chinese pages. Chances are great that a site with parallel text have such links in both directions. This fact is used as the criterion of candidate sites search.

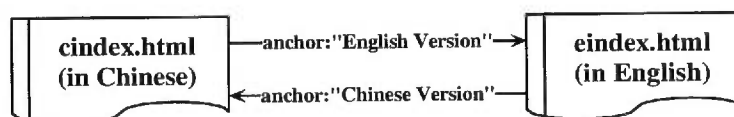


Figure 2.2: A pair of parallel pages linking to each other.

Fig. 2.3 depicts the process of candidate sites searching. Taking advantage of the

---

<sup>2</sup>An anchor text is a piece of text in a Web page which you can click to go to where it links to. To be instructive, it usually contains the key information of the linked page.

keyword searching and language identification functions of AltaVista, we can send a query in the form

*anchor : "english version" ["in english", ...]*

and set the language option as Chinese for pages in Chinese. Then we send another query in the form

*anchor : "chinese version" ["in chinese", ...]*

and set the language option as English for pages in English.

From the two sets of pages obtained by the above queries we extract two sets of Web sites. The intersection of these two sets is then the candidate sites with links in both directions. Considering that search engines only index part of all the pages of one site, we take the union of the two sets instead of the intersection. That is to say, a site is a candidate site when it is found to have either an English page linking to its Chinese version or a Chinese page linking to its English version. With this loose criterion we could reduce the loss because of the incompleteness of the search engines.

### 2.2.3 File Name Fetching

It is our assumption that a pair of parallel texts exists in the same site. To search for parallel pairs from a site, PTMiner has to first obtain all (or at least part of) the HTML file names of the site. From these names, pairs are scanned. The amount of file names we obtain from each site directly affects the amount of the pairs we can find. As the first step of collecting file names, we query the search engines in the form

*host : www.info.gov.hk*

to fetch the Web pages that they indexed from each site.

It is more efficient to fetch file names from search engines than to crawl the sites themselves. However, obviously search engines only index part of the pages of each candidate site. Another problem is that some search engine like AltaVista has a limit

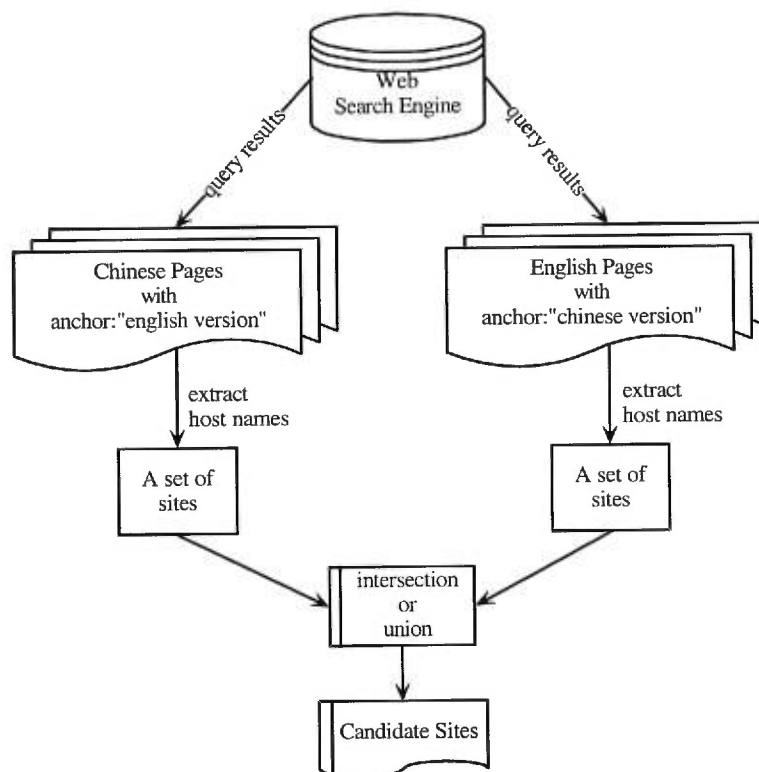


Figure 2.3: Candidate sites search using Web search engines.

on the search results that a user can obtain. One can only view 1000 results of each query even though AltaVista has a lot more. For more results we send two queries to AltaVista instead of one. Each query asks for the pages in one language of the language pair. For example, one query for pages in English and the other for pages in Chinese. In this way we could double the limit of results. For large sites from which AltaVista indexed thousands of pages, a lot of pages still cannot be reached. Therefore, the next step, host crawling, is needed.

#### 2.2.4 Host Crawling

The previous step using the existing search engines may find sufficient number of file names for further searching if, for example, there are a great number of candidate sites for some language pairs such as English-French. For language pairs such as English-Chinese, there are fewer candidate sites. Therefore, a more thorough search



is required in each candidate site in order to find as many parallel texts as possible from it. For this purpose, we built a host crawler, which is similar to Web crawlers to explore more thoroughly each candidate site. Web crawlers are used by search engines to index Web pages. They go through innumerable pages and hosts on the Web. A host crawler is a Web crawler for a single host. It does not follow links to documents outside the host.

A breadth-first crawling algorithm starting from a initial set of file names (result of file name fetching) is applied in the host crawler of PTMiner. We describe the algorithm in the following pseudo-code.

```
public class HostCrawler {
    private HashSet fileSet = new HashSet(); // New empty set
    private int max = 10000; // Maximum number of URLs to crawl

    public HostCrawler(HashSet iniSet) {
        fileSet.add(iniSet);
        crawl_set(iniSet);
    }

    private void crawl_set(HashSet rootSet) {
        HashSet thisSet = new HashSet();
        // Crawl each URL in rootSet and put newly found URLs into thisSet
        while (rootSet.hasMoreURLs() && fileSet.size() <= max)
            crawl(rootSet.nextURL(), thisSet);
        // If there are newly found URLs (in thisSet), recursively crawl thisSet
        if ((! thisSet.isEmpty()) && (fileSet.size() <= max))
            crawl_set(thisSet);
    }

    private void crawl(URL root_url, HashSet thisSet) {
        open root_url;
        while (reading through root_url) {
            if ((found new URL) && (new_url is not in fileSet)) {
                thisSet.add(new_url);
                fileSet.add(new_url);
            }
        }
    }
}
```

Given enough time, a host crawler is supposed to crawl out all the file names that are linked (directly or indirectly) from the initial set of pages. It is yet time-

consuming. It took 3-4 days to find 55971 file names from *www.info.gov.hk*, the information site of Hong Kong government. Whether it is worth applying depends on the amount of text demanded and whether we can satisfy the demand without applying it. It is necessary in our mining for Chinese-English parallel texts.

### 2.2.5 Pair Scan (by Naming Patterns)

After collecting file names for each candidate site, the task left is to find out parallel pairs from them. A straightforward method seems to be comparing every couple of files. It is not applicable because of the following reasons:

- 1 The complexity of this algorithm has a quadratic order in terms of the number of the files. When we have to process thousands of files for each site, the computing time is not affordable.
- 2 All the files have to be downloaded locally to be processed. It gives a high load on the network and local file system. Needless to say, much time will be wasted for downloading non parallel texts.

We can divide the characteristics of a Web page into two classes, external features and internal features. By external feature, we refer to the features that may be known without analyzing the contents of the file such as its URL, size, and date of modification. The internal features such as language, character set, and HTML structure cannot be known until we download a page and analyze its contents.

Instead of directly looking into the internal features of the candidate files, we use a heuristic method which scans only by an external feature: naming patterns that are usually found in parallel Web pages.

The scanning criterion comes from the following observation: parallel text pairs usually have similar name patterns. The difference between the names of two parallel pages usually lies in a segment which indicates the language. For example, “file-ch.html” (in Chinese) vs. “file-en.html” (in English). The difference may also appear in the path, such as “.../chinese/.../file.html” vs. “.../english/.../file.html”. Fig. 2.4

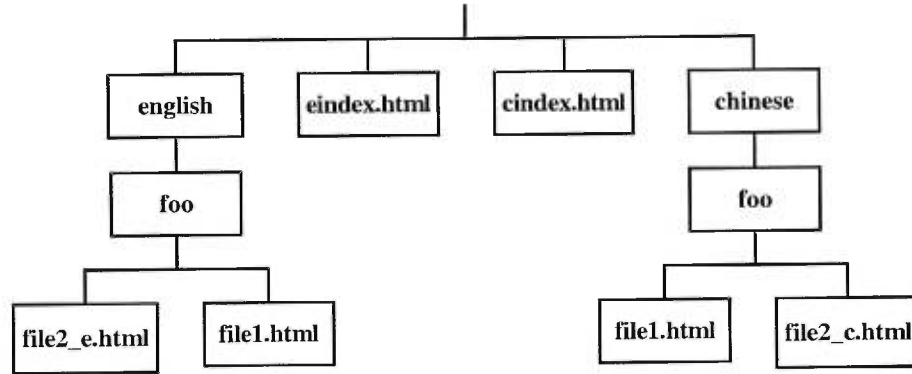


Figure 2.4: Parallel pairs in a directory tree.

shows some possible ways parallel pairs exit. The name patterns described above are commonly used by the webmasters to help organizing their sites. Hence we can suppose that a pair of pages with this kind of pattern are most probably a pair of parallel pages.

The general idea of the pair scan algorithm is as follows. For each file name, construct new file names that could become a pair together with the current file and check if such a file really exists.

More specifically, we establish four arrays for English prefixes, English suffixes, Chinese prefixes and Chinese suffixes as listed below.

$$\begin{aligned} \text{English Prefix} = \{ & "e", "en", "eng", "engl", "english", \\ & "e-", "en-", "eng-", "engl-", "english-", \\ & "e-", "en-", "eng-", "engl-", "english-" \} \end{aligned}$$

$$\begin{aligned} \text{Chinese Prefix} = \{ & "c", "ch", "chi", "chin", "chinese", \\ & "c-", "ch-", "chi-", "chin-", "chinese-", \\ & "c-", "ch-", "chi-", "chin-", "chinese-" \} \end{aligned}$$

$$\begin{aligned} \text{English Suffix} = \{ & "e", "en", "eng", "engl", "english", \\ & "_e", "_en", "_eng", "_engl", "_english", \\ & "-e", "-en", "-eng", "-engl", "-english" \} \end{aligned}$$

$$\begin{aligned} \text{Chinese Suffix} = \{ & "c", "ch", "chi", "chin", "chinese", \\ & "_c", "_ch", "_chi", "_chin", "_chinese", \\ & "-c", "-ch", "-chi", "-chin", "-chinese" \} \end{aligned}$$

For each file, the following tasks are carried out:

- 1 Extract the path and the file name;
- 2 Add all the prefixes and suffixes to the file name (once a time) to see if there is such a file in the same path;
- 3 If the file starts (ends) with some prefix (suffix), replace it with its correspondence in the other language. For example, replace "en-" for "ch-". Then check if there is such a file in the same path among the file names we found previously; (An alternative is to send a query to the candidate site to try to locate the file. This is however more time-consuming. Therefore, it is not adopted.)
- 4 For each directory level of the file's path:
  - 4.1 Change the directory name as in step 2 and 3 to form new paths;
  - 4.2 In each new path, try to locate the original file name or its variations produced as in step 2 and 3.

In this algorithm, many variations of each file name are checked. However, the computing time for each file is a constant. The whole processing time increases linearly with the number of the files.

In this step we search for possible parallel pairs only based on file name and path. The results proved that the accuracy of this approach is acceptable (see Section 2.4).

### 2.2.6 Filtering

We then compare the two files of each pair by some other external or internal features. We now describe some methods that we used.

#### Text Length

Apparently a good parallel pair usually have similar file lengths. The simplest way is then to compare the lengths of the two files. The only problem is to set a reasonable

threshold that can filter out mostly wrong pairs without sacrificing too many good ones, i.e., balance between recall and precision. The usual difference ratio depends on which language pair we are dealing with. For example, Chinese-English parallel texts usually have higher difference ratio in length than that of English-French parallel texts.

The filtering threshold has to be set from the actual observations. We will give detailed analysis in section 2.4.

### **Language and Character Set**

It is also obvious that the two files of a pair have to be in the two languages. By identifying language and character set, we can filter out the pairs that do not satisfy this basic criterion. Some Web pages explicitly indicate the language and the character set. More often such information is omitted by author. We need some language identification tool for this task.

The SILC system [IFP97] is a language and coding identification system developed by the RALI laboratory of University of Montreal. It “possesses, for each of the language and coding pairs it handles, a model that assigns a certain probability to the text in question. The system also incorporates formal criteria that allow it to select the model that obtains the best score.” [IFP97] As we shall see, the precision of SILC is good enough to be used to eliminate wrong pairs.

In PTMiner system, we actually use length and language as the two criteria in the filtering process. These two simple methods turned out to be effective, as we will see in the mining result analysis in section 2.4.

### **HTML Structure and Alignment**

In the STRAND system [Res98], the candidate pairs are evaluated by aligning them according to their HTML structures and computing confidence values. Pairs are supposed to be wrong if they have too many mismatching markups or low confidence values.

Comparing HTML structures seems to be a sound way to evaluate candidate pairs since parallel pairs usually have similar HTML structures and thus similar appearance. However, we also noticed that differences may exist. Sometimes the difference is large while the evaluated pair is parallel. This is particularly the case for English and Chinese because the author may use two different Web page editors for the two languages. Caution has to be taken when measuring structure difference numerically.

Parallel text alignment is still an experimental area. The measurement of confidence value of alignment is even more complicated. For example, the alignment algorithm we used in the training of the statistical translation model gives acceptable alignment results but no confidence value that we can “confidently” use as an evaluation criterion. In the current version, these two criteria are not used for filtering purpose. However, it is relatively easy to add them later on. In fact, two criteria are implicitly used in a later process – sentence alignment before the training of translation model. In sentence alignment, we use HTML markups as cognates, and we only select 1-1 alignments (the kind of alignment that is the most reliable) to train the model. Therefore, the above two criteria will be partially used in the whole process.

## 2.3 PTMiner – A Multi-Tier Distributed Text Miner

As our implementation of parallel text mining tool, PTMiner is a multi-tier distributed Web search agent. It is designed to be effective, efficient, scalable, easy to maintain, easy to analyze mining results, and easy to modify for other language pairs. We explain its features in the following.

- 1 **Efficiency**     The mining process could take long time, especially when host crawler is used. By adopting a distributed model, PTMiner can process several candidate sites in parallel and thus reduce greatly the processing time.
- 2 **Scalability**     If needed, PTMiner can be used to generate very large parallel corpus provided that there are enough such parallel texts on the Web. If a small-size corpus is demanded, it can produce it with only the information from

Web search engines (without using host crawler) in a short time. Therefore, we determine whether to use host crawling according to the actual need.

- 3 **Platform Independence** Taking advantage of the Java technology, most modules of PTMiner can run in various systems, Solaris, Linux or Windows NT. It provides the flexibility on using system resources.
- 4 **Maintenance** PTMiner is a distributed system that involves various processes in various machines. It is however easy to start and restart the system. A centralized monitoring GUI interface is provided for the user to watch clearly the working situation of all the processes, the content of the PTMiner database as well as the overall mining progress. Because the intermediate information is stored in the database and file system, the system can restart from where it was stopped without wasting time to repeat the same work.
- 5 **Result Analysis** The meta-information of the mining results is stored in the PTMiner database which greatly facilitates the result analysis. User can get all kinds of statistical data by SQL queries.
- 6 **Adaptability** The current implementation is designated to Chinese-English parallel text mining. However the general idea behind the algorithm of PTMiner is language independent. Thus the system can be adapted to the mining of other language pairs with only some minor modifications such as naming patterns and queries to search engines.

### 2.3.1 Adopted Technologies

Before explaining the details of the architecture and implementation of the PTMiner system, we would first give a brief introduction on distributed object technologies and the JDBC technology which are adopted in PTMiner and have made the design objectives possible.

## Distributed Object Technologies

Combining networking and object-oriented programming (OOP) technologies, distributed object computing are extensively applied in business software for large organizations. By properly distributing objects designated to various tasks in network, distributed object systems provides coherent, effective and efficient performance.

With its polymorphism, inheritance, and encapsulation, object-oriented programming is described to provide two important properties: “interchangeability and interoperability” [MCD98]. When OOP is applied in distributed systems, these two properties are greatly enhanced by the unique advantages of networks. Objects can be more freely exchanged through network. More importantly, collaborating together, objects distributed in networks offer effective solution to geographically dispersed computing tasks.

“A distributed object technology aims at location transparency, thus making it just as easy to access and use an object on a remote node (called, logically enough, a remote object) as an object on a local node” [MCD98]. Among other functions, a distributed object technology should at least enable locating and referencing remote objects as well as remote method calls.

The current dominant distributed object technologies are SUN’s Remote Method Invocation (Java RMI), Microsoft’s Distributed Component Object Model (DCOM), Object Management Group’s Common Object Request Broker Architecture (CORBA), and ObjectSpace’s Voyager. Details of these technologies are beyond the scope of this thesis. The book, *Java Distributed Objects* [MCD98], is a good reference introducing and comparing these technologies.

CORBA is a technology that enables applications throughout networks to communicate with each other no matter where they are located or by which language they are implemented. Developers specify interfaces of their applications with Interface Definition Language (IDL). Remote method invocations are realized by Object Request Brokers (ORB) communicating using a language-independent protocol, Internet Inter-ORB protocol (IIOP). ORB acts as the middle-ware that seamlessly establishes



client-server connections (including finding an object, passing parameters, invoking methods, and returning results) between dispersed objects. The ORB in the Voyager package of ObjectSpace was used to implement two servers (Crawler and Scanner) in the PTMiner system.<sup>3</sup>

## JDBC

Database connection is involved in a large proportion of applications. The idea behind both Microsoft's ODBC (Open Database Connectivity) and JDBC (Java Database Connectivity) is to provide standard APIs that enable developers to write almost the same program to access virtually any relational databases. Specifics of connecting and querying a certain database are hidden in the driver for the database and thus transparent to developers.

The JDBC technology consists of a Java API and a series of drivers implemented for different databases. With the JDBC API and corresponding drivers, developers could build platform-independent database applications with little vendor-specific knowledge. On the other hand, it is still possible to make vendor-specific call through JDBC. Users are not restricted by the standard.<sup>4</sup>

In PTMiner, the database is built with the MySQL server. The other modules access the database through JDBC connection.

### 2.3.2 PTMiner Architecture and Implementation

Fig. 2.5 illustrates the system architecture of PTMiner. Arrows indicates the directions of data flow between modules. It also shows how modules communicate with each other (through JDBC connection, CORBA remote method call or UDP (User Datagram Protocol) packet).

---

<sup>3</sup>For more information about CORBA and Voyager, see <http://www.omg.org> and <http://www.objectspace.com>.

<sup>4</sup>For more information about JDBC, see <http://java.sun.com/products/jdk/1.2/docs/guide/jdbc/index.html>.

Briefly speaking, the central control unit of the system is the PTMiner server which reads candidate sites from the database and assign them to Crawler and Scanner servers. Crawlers and Scanners reside in different machines. They register in the database when starting. Each site has to be passed to a Crawler server to collect file names of this site, and then a Scanner server to scan for parallel pairs. Important results are stored into the database. The PTMiner server relies on the database to synchronize Crawlers and Scanners. PTMonitor is a central GUI interface receiving messages (in UDP packets) from all servers. It is also a viewer of the database content.

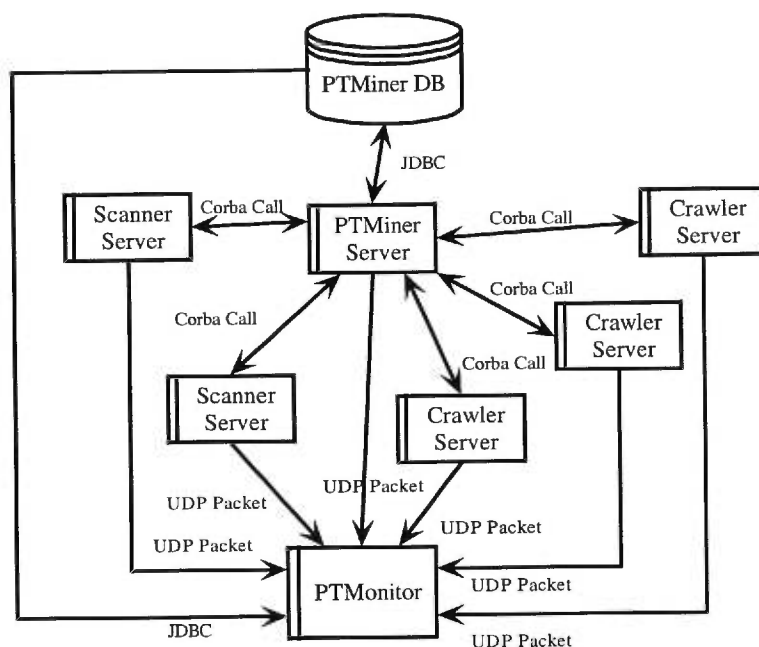


Figure 2.5: The architecture of PTMiner.

We now describe each module in detail.

### PTMiner Database

The PTMiner database serves as the storage of intermediate and final mining results as well as working situation of the servers. Fig. 2.6 is the entity-relationship diagram (ERD) of the database. The entity *site* represents candidate sites with *host* (host name) as the unique key. Its attributes *searched*, *crawled*, and *scanned* indicates if the site has been searched (from search engines), crawled, or scanned, respectively.

Corresponding numbers are stored in the attributes *searched pages*, *crawled pages*, and *scanned pairs*. When a site has been completely processed, the *url* and *host* of each file of all the pairs found will be stored in the *file* table. After the files are downloaded and their language and character set are determined, the other attributes, *length*, *language*, and *charSet*, are filled. With statistical analysis, we decide the criteria that judge if each *pair* relationship is true.

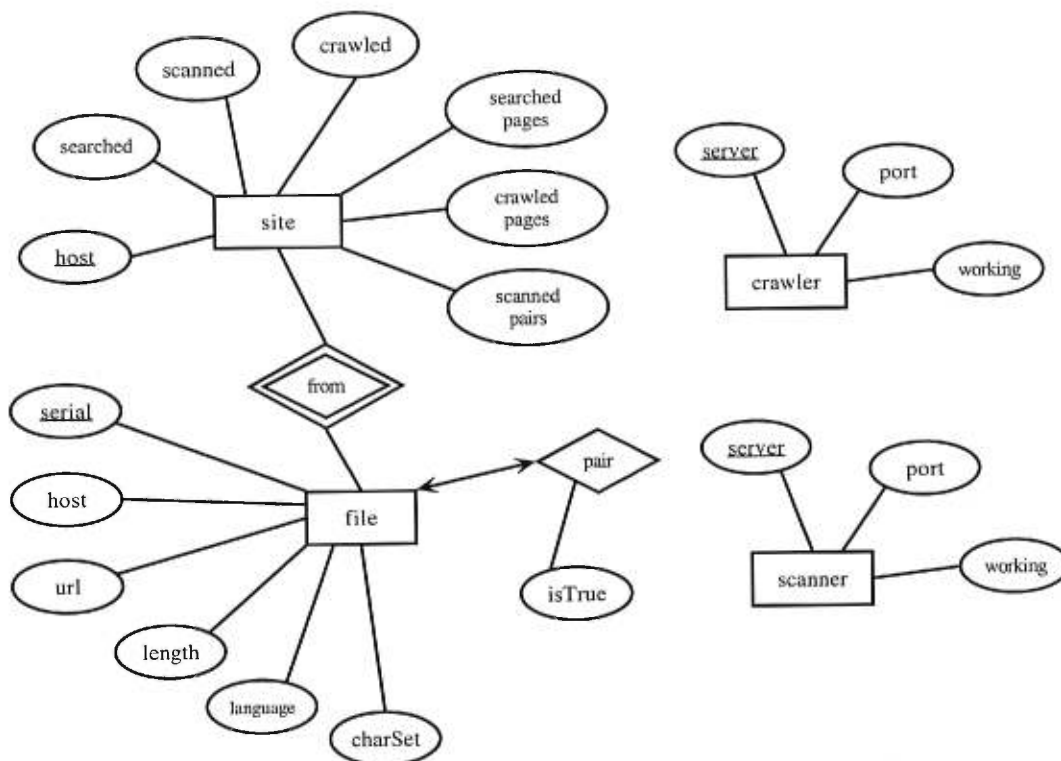


Figure 2.6: The ERD of the PTMiner database.

The *crawler* and *scanner* tables stores information of available crawler and scanner servers, i.e., their *server* name, *port*, and *working* status. Note that being used only for helping the mining process, this database is not completely normalized as most relational databases.

The database is implemented with MySQL, a multi-threaded SQL database server. MySQL is a fast, robust, and easy-to-use database server which fits the needs of PTMiner very well. It can handle very large databases with very good performance. It is multi-threaded which makes parallel processing possible. Its functionality is

much simpler than that of the large commercial DBMSs such as ORACLE, DB2 or SYBASE. However, MySQL provides enough functions that the current PTMiner requires as well as the ease of use.<sup>5</sup>

### Candidate Site Fetcher

The candidate site fetcher module is a stand-alone program (not shown in Fig. 2.5) which implements the first step, candidate sites search, of the mining algorithm. It sends queries as mentioned in sub-section 2.2.2 to AltaVista and retrieves a set of candidate sites. Only AltaVista is used because its ability of identifying Chinese Web page. The sites are stored into the database for future processing.

### Crawler Server

The Crawler server is a CORBA server providing the following interface (in IDL):

```
interface ICrawler {
    boolean isWorking();
    void stopWorking();
    long fetch(in string site);
    long crawl(in string site);
};
```

When a Crawler server is started, it first registers in the database and also notifies PTMonitor. Then it waits for invocation from the PTMiner server. The *fetch* method takes the name of a candidate site and fetches file names from AltaVista and Northern Light. The result file name log will be read by the *crawl* method as the initial set for the host crawler. Messages showing progress or exceptions encountered are sent to PTMonitor in UDP packets. The numbers of the fetched and crawled file names will be returned to the PTMiner. The host crawling algorithm was described in sub-section 2.2.4.

### Scanner Server

Also a CORBA server, the Scanner server has the following interface:

---

<sup>5</sup>For more information about MySQL, see <http://www.mysql.com>.

```
interface IScanner {
    boolean isWorking();
    void stopWorking();
    long scan(in string site);
};
```

Similarly to the Crawler server, the Scanner server registers itself in the database and sends messages to PTMonitor. The *scan* method takes a site name, opens the corresponding file name log, and then scans for parallel pairs by naming patterns. The number of pairs found is returned to the PTMiner server.

### PTMiner Server

As stated above, the PTMiner server is the central control unit of the system. It synchronizes the real workers, Crawler servers and Scanner servers, according to information in the database. The following pseudo-code shows how the server works.

```
public class PTMiner {
    main () {
        while (true) {
            query the database for unfinished candidate sites;
            if none, exit;

            query the database for uncrawled sites;
            query the database for available Crawlers;
            if (there are uncrawled sites and available Crawlers)
                open a Crawl_Thread for each uncrawled site and available Crawler;

            query the database for unscanned sites;
            query the database for available Scanners;
            if (there are unscanned sites and available Scanners)
                open a Scan_Thread for each unscanned site and available Scanner;
        }
    }
}

public class Crawl_Thread {
    public CrawlThread (String site, String crawlServerName,
                       int crawlServerPort) {
        locate the Crawler server and reference it as thisCrawler;

        query the database to check if the site has been searched;
        if (the site has not been searched) {
```

```

    // remote method call
    int searched_pages = thisCrawler.fetch(site);
    update site.searched and site.searched_pages in the database;
}

// remote method call
int crawled_pages = thisCrawler.crawl(site);
update site.crawled and site.crawled_pages in the database;

}
}

public class Scan_Thread {
    public ScanThread (String site, String scanServerName,
        int scanServerPort) {
        locate the Scanner server and reference it as thisScanner;

        // remote method call
        int scanned_pairs = thisScanner.scan(site);
        update site.scanned and site.scanned_pages in the database;
        insert the pairs newly scanned into the file table and
            the pair table in the database;
    }
}
}

```

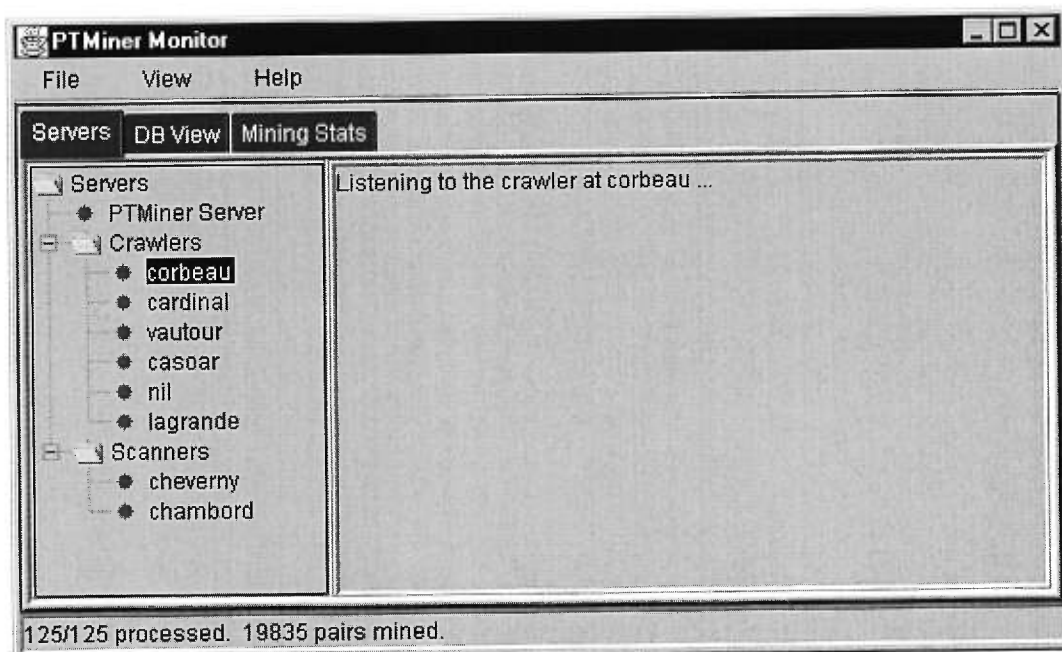
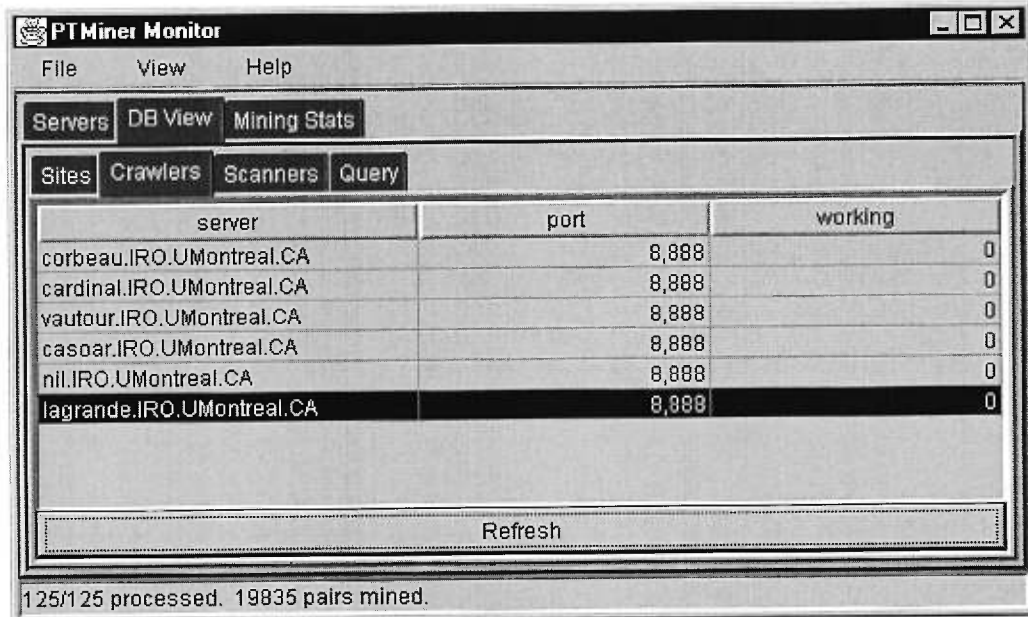


Figure 2.7: PTMonitor showing messages from the servers.

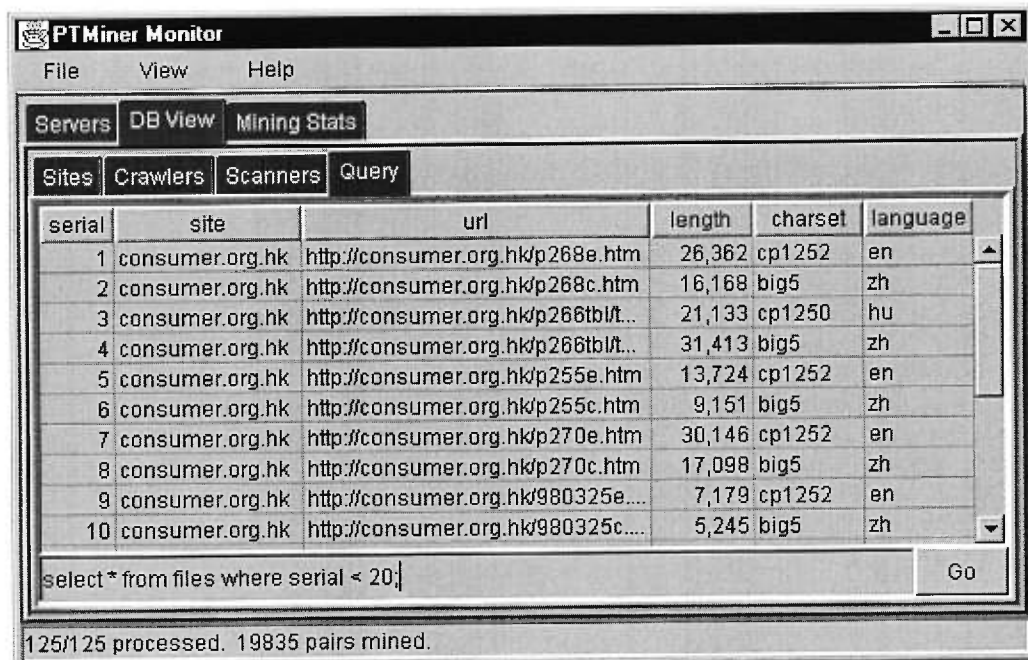


server	port	working
corbeau.IRO.UMontreal.CA	8,888	0
cardinal.IRO.UMontreal.CA	8,888	0
vautour.IRO.UMontreal.CA	8,888	0
casoar.IRO.UMontreal.CA	8,888	0
nil.IRO.UMontreal.CA	8,888	0
lagrande.IRO.UMontreal.CA	8,888	0

Refresh

125/125 processed. 19835 pairs mined.

Figure 2.8: PTMonitor showing the crawler table of the database.



serial	site	url	length	charset	language
1	consumer.org.hk	http://consumer.org.hk/p268e.htm	26,362	cp1252	en
2	consumer.org.hk	http://consumer.org.hk/p268c.htm	16,168	big5	zh
3	consumer.org.hk	http://consumer.org.hk/p266tb/t...	21,133	cp1250	hu
4	consumer.org.hk	http://consumer.org.hk/p266tb/t...	31,413	big5	zh
5	consumer.org.hk	http://consumer.org.hk/p255e.htm	13,724	cp1252	en
6	consumer.org.hk	http://consumer.org.hk/p255c.htm	9,151	big5	zh
7	consumer.org.hk	http://consumer.org.hk/p270e.htm	30,146	cp1252	en
8	consumer.org.hk	http://consumer.org.hk/p270c.htm	17,098	big5	zh
9	consumer.org.hk	http://consumer.org.hk/980325e....	7,179	cp1252	en
10	consumer.org.hk	http://consumer.org.hk/980325c....	5,245	big5	zh

select \* from files where serial < 20; Go

125/125 processed. 19835 pairs mined.

Figure 2.9: PTMonitor showing the result set of the user defined query.

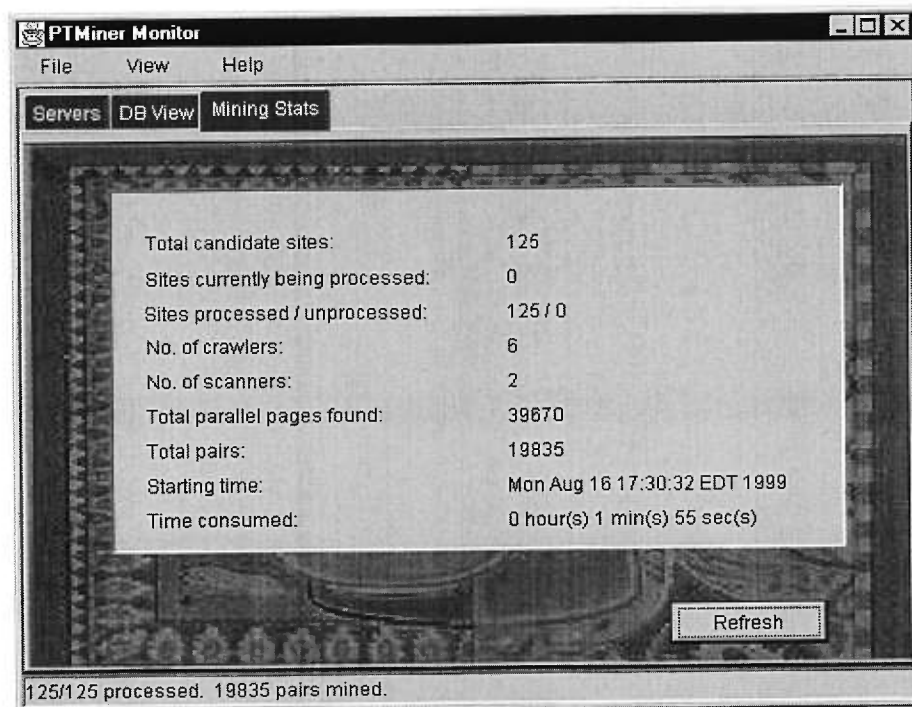


Figure 2.10: PTMonitor showing mining progress.

## PTMonitor

PTMonitor is a GUI interface implemented with Java Swing API. The objective is to facilitate the monitoring of the whole mining process. It presents various kinds of information including:

- Messages from Crawler servers, Scanner servers, and the PTMiner server – As shown by Fig. 2.7, each server is represented by a node in the tree and each node has an associated text area. PTMonitor listens at some port for UDP packets from the servers. Once a packet has arrived, it determines where it is from and appends the message at the corresponding text area.
- Contents of the database – With JDBC connection, PTMonitor reads from the database and display contents of the *site*, *crawlers* (Fig. 2.8), and *scanners* tables. It also gives users the freedom of sending new queries (Fig. 2.9).



- Statistical data – As shown in Fig. 2.10, PTMonitor also shows some data giving an overview to the mining progress.

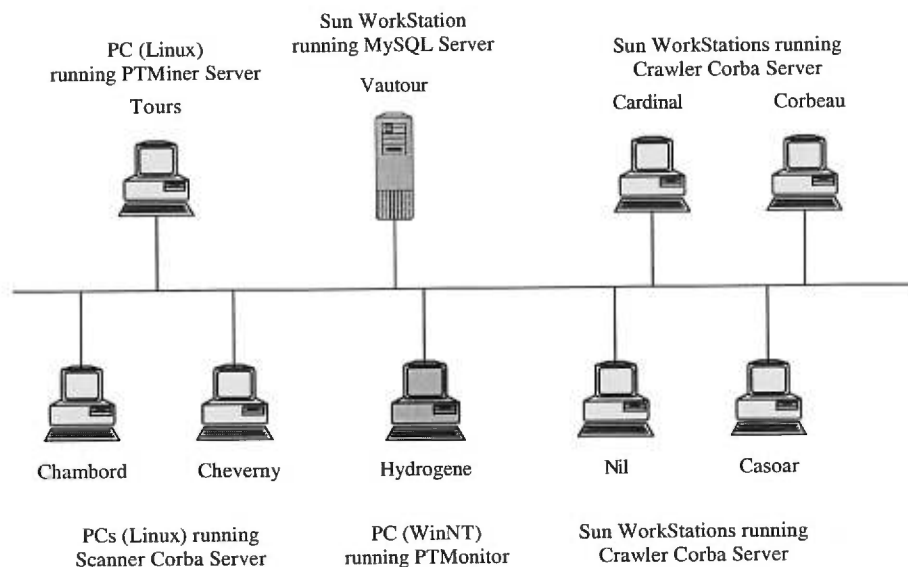


Figure 2.11: A network view of the PTMiner system.

### Allocating System Resource

One advantage of the PTMiner system is that most of its modules are implemented in Java which enables them to run on practically any platform. This feature brings convenience in distributing working objects. Most modules of PTMiner such as the MySQL server, the PTMiner server, and the Crawler servers consumes very few (around 1% or less) percentage of CPU time. Thus they could be established in any machine without influencing other users. The only module that costs most CPU time is the Scanner server. Fortunately, its actual working time on each site is much shorter than that of Crawler servers. We may need many Crawler servers but only one or two Scanner servers. Fig. 2.11 is an example of the working situation of PTMiner.

## 2.4 Mining Results Analysis

The PTMiner system has been formally run twice. We limited the candidate sites at the domain *hk* because Hong Kong is a perfect Chinese-English bilingual city with high possibility of having high quality parallel sites (another nice domain may be Singapore). The first run found 5276 pairs (before filtering) from 196 candidate sites in several hours without applying the host crawler. After filtering, 3316 pairs were identified as true parallel pairs.

The second run was carried out after the system was completely implemented. Host crawler was included to collect file names from candidate sites exhaustively. To limit the running time, the maximum number of file names to crawl was set at 10000 per site. It took around one week to obtain 19835 pairs (searched only by naming patterns) of parallel pages from 185 candidate sites. Less candidates were found this time because the contents of AltaVista changed. 14820 pairs were left after applying the filtering criteria on text length and language. The resulted corpus has 117.2M Chinese text and 136.5M English text.

It is now necessary to evaluate the quality of the generated corpus. In the following we will present the estimated mining precision before and after filtering as well as how the filtering criteria were decided.

### Manual Evaluation on Sample Pairs

To estimate the precision of the unfiltered pairs, we randomly picked 367 pairs and observed each of them. Among the 367 pairs, 301 pairs were found to be true parallel pages. That is a precision of 82%. Analyzing the bad pairs, we found that they include:

- 1 Incorrect URLs. It may be because the pages are out-dated but still indexed by the search engines.
- 2 Pages that are designed to be parallel yet not good enough. Their contents cannot match well with each other.

3 Pages that are good parallel pairs yet consisted of mostly graphics instead of text. They are nice parallel pages but not parallel text.

4 Pairs that are totally not parallel. Their file names happen to match the naming rules.

Next we determine the filtering criteria according to the analysis of the sample pairs.

### Language and Character Set

When detecting language and character set, we rip off the HTML markups from pages to eliminate their effect on SILC. The criterion seems to be straightforward. A pair has to have a page in English and the other page in Chinese. However, the precision of SILC is not 100%. We have to tune the criterion according to SILC's mistakes.

In the total 39670 mined pages, SILC detected 18041 Chinese pages, 425 Korean pages and 183 Japanese pages. Through observation we found that most Korean and Japanese pages are actually in Chinese and mistaken by SILC. We suppose the cause is the similarity between CJK character sets. Hence we regard all the pages detected in these three languages as Chinese pages. This decision is safe because we restricted the mining in HK domain.

SILC also found 13351 pages are in English and 5444 pages are in other European languages such as French, Hungarian, or Spanish. The same phenomenon as above was observed. Therefore, we take all these pages as in English.

### Text Length

It is our intuition that parallel pages tend to have similar lengths. Questions are what the normal differences are and which threshold can be taken as the filtering criterion. First of all, we define the length difference as

$$length\_diff = \frac{|file1.length - file2.length|}{\min(file1.length, file2.length)}.$$

We calculated the length differences of the 301 true parallel pairs in the samples. Fig. 2.12 shows their distribution. We can see that there is a big drop between *length\_diff* 40% and 50%.

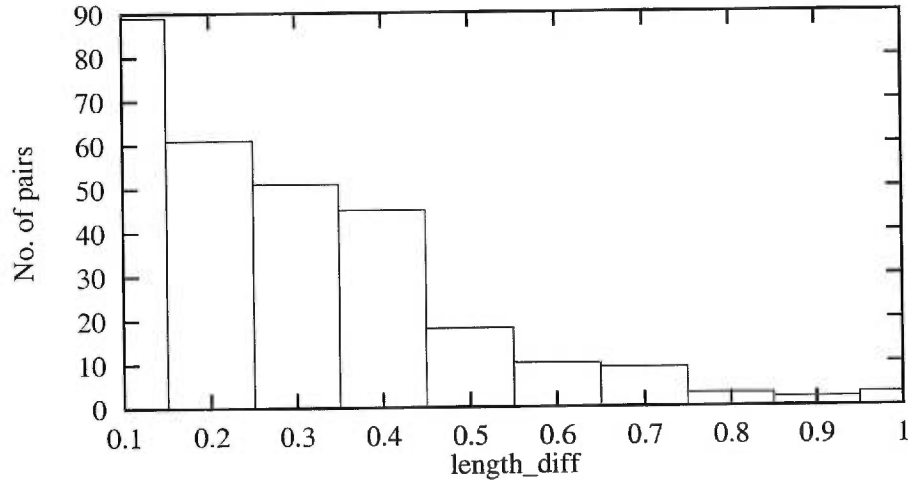


Figure 2.12: The distribution of true pairs on different length differences.

We then apply different values as the criterion of *length\_diff* to filter the manually evaluated samples and record the obtained precision (Fig. 2.13). We get the maximum precision, 85.7%, when the criterion is 40%. Should we then use 40% as the threshold? The other side of the problem, recall ratio, raises itself. Obtaining the maximum precision leads to a large loss of recall ratio.

For example, in the whole mined pairs, there are 12883 pairs with length difference less than 40% and 16984 pairs with length difference less than 100%. From Fig. 2.13, the precision difference between using 40% and 100% as the threshold is about 1%. That is to say, If we use 40% instead of 100% as the criterion, we may get 1% more precision by filtering out 4101 more pairs. However, in the 4101 pairs there may still exist a considerable amount of true pairs. We wouldn't lose many potential true pairs for a little more precision.

On the other hand, filtering by language detection has better effect than filtering by text length. It can improve the precision significantly without sacrificing the recall ratio. For example, if we set the criterion of length difference at 100% and also use

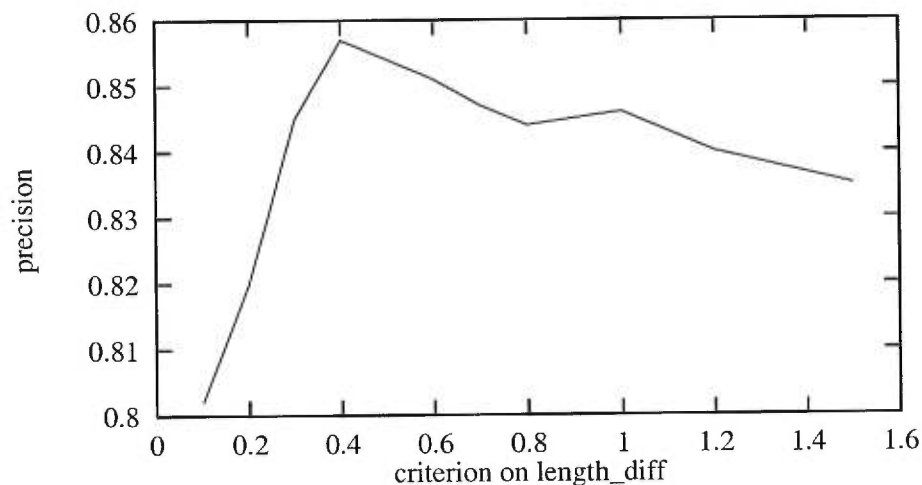


Figure 2.13: Precision obtained applying different criteria on length difference.

language detection, we get a precision of 93% on the sample pairs, which means an improvement of 11%.

From the above analysis, we loose the threshold in length difference to 120% and use the language identification criterion as described by the following SQL query:

```
SELECT *
FROM files as file1, files as file2, pairs
WHERE pairs.file1 = file1.serial and pairs.file2 = file2.serial
and file1.length > 0 and file2.length > 0
and
(abs(file1.length - file2.length)/least(file1.length, file2.length) < 1.2)
and
(((file1.language = 'zh'
  or file1.language = 'ko'
  or file1.language = 'ja')
and
(file2.language != 'zh'
and file2.language != 'ko'
and file2.language != 'ja'))
or
((file2.language = 'zh'
  or file2.language = 'ko'
  or file2.language = 'ja')
and
(file1.language != 'zh'
and file1.language != 'ko'
and file1.language != 'ja'))))
```

The query resulted in 14820 pairs with the precision that we estimate at around 90%. These pairs will be used in the training of the translation model.

## 2.5 Summary

In this chapter, we reviewed some important knowledge discovering technologies such as OLAP, data mining, text mining and Web mining. We then introduced the algorithm and implementation of the PTMiner system, which is an intelligent Web search agent designed for large-scale parallel text mining. The mining algorithm consists of candidate sites search, file name fetching (host crawling) and pair scan by naming patterns. Without having to look into the internal contents of documents, it is language independent. The distributed implementation model of PTMiner enables it to find large amount of parallel text in a relatively short time. We also explained how we used text length difference and language identification technique to evaluate the mined pairs and thus improve mining precision.

## Chapter 3

# Training the English-Chinese Statistical Translation Model

As the second step of our work, we try to establish an English-Chinese statistical translation model trained by the parallel text we collected from the Web. (The RALI group has developed tools to build translation models). What we obtained in the last step were 14820 pairs of presumably parallel HTML pages. The input to the training process of the translation model has to be aligned sentences. Thus a series of pre-training processing has to be done on the raw texts before they can be really used to train the translation model. Among others, sentence separation, parallel text alignment, Chinese segmentation and English expression extraction are vital to the performance of the final translation model.

Fig. 3.1 depicts the pre-training procedures. First of all, a pair of HTML files are divided into sentences according to not only the punctuation, but also their HTML structures. The two markupt files, *e.cesana* and *c.cesana* are then passed to the alignment program. The alignment algorithm considers both sentence lengths and HTML structures. The file, *src.al*, stores the alignment data. At the same time, we strip the HTML markups off from the English and Chinese sentences. English words have to be converted to their citation forms. According to a dictionary, English expressions are extracted and appended to the sentences they belong to. For

Chinese, all the texts have to be converted to the same code (GB). Unlike English, Chinese sentences are constructed by characters with no space explicitly indicating word boundaries. Chinese segmentation is then essential. Fig. 3.2 shows what the final input files of the training look like.

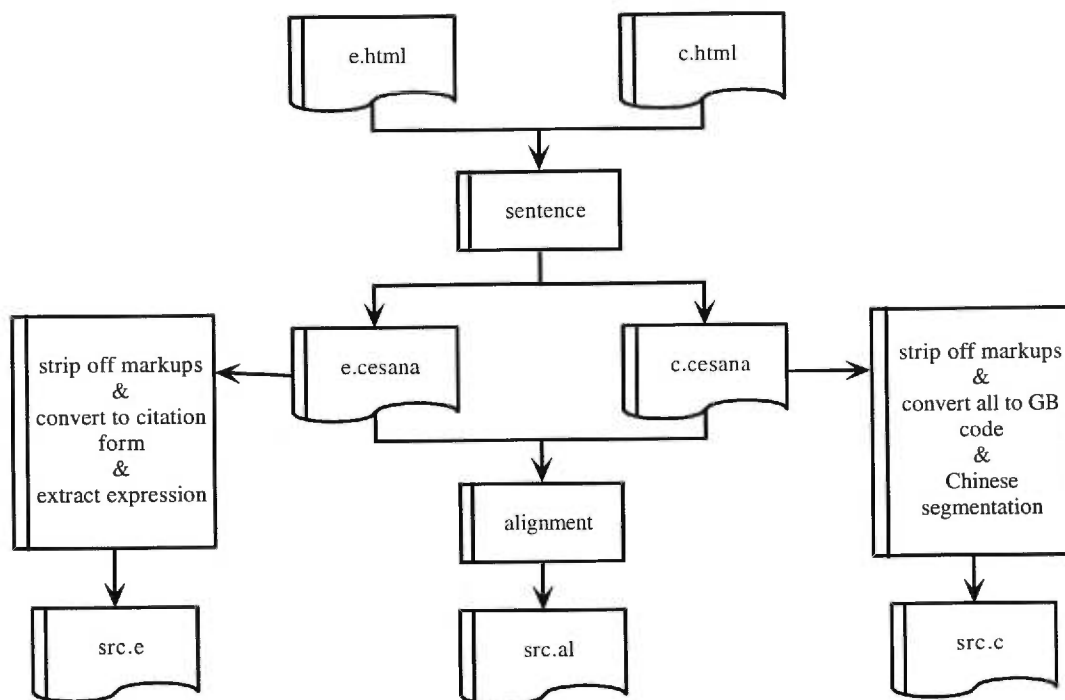


Figure 3.1: Pre-training procedures.

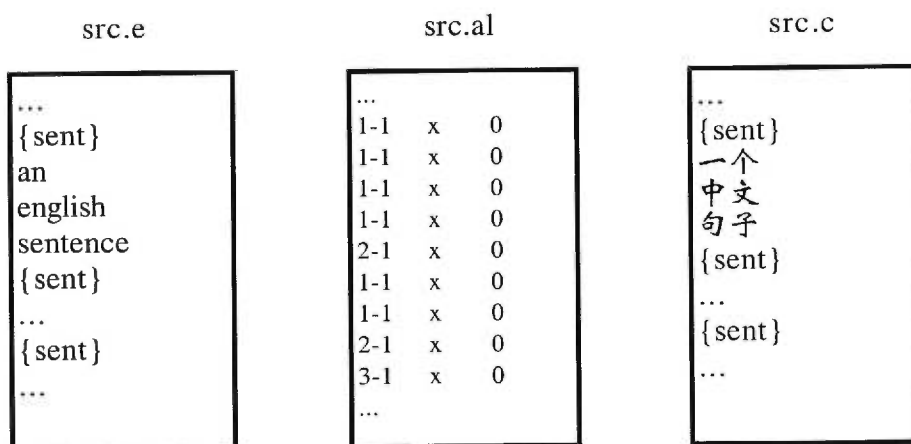


Figure 3.2: Sentences are assembled as the training source.

In the following sections, we will first introduce some related Chinese information



processing techniques. The details of text alignment, Chinese segmentation, and English expression extraction will then be explained. Afterwards, we shall go through the principles of the statistical translation model. The first straightforward application of the trained model would be an English-Chinese lexicon with which we examine the precision of the model. We will finally present these results.

## 3.1 Related Chinese Information Processing Techniques

Computer was capable of processing only English (or European languages) information at the beginning. With the evolution of computer science, computers are expected to deal with all languages as needed, including oriental languages such as Chinese, Japanese and Korean (CJK), which have much more complex writing systems than those of western languages. Information processing of CJK languages differs from western languages in their writing systems, character set standards, encoding methods, input methods and typography.

Chinese information processing techniques are inevitably involved in this project. In this section, we will introduce some essential concepts, standards and techniques that are related to this work.

### 3.1.1 Coded Chinese Character Set Standards

Unlike the western languages which have handful sets of characters such as English alphabet, Chinese has tens of thousands characters. Thousands of them are frequently used while others are not. It is important to define a set of most important characters for educational or information processing purpose. A character set specifies a set of characters (out of all the characters of the language) that are the most important for communication. A coded character set standard is a character set that is encoded for computer processing.

Two Chinese character set standards, GB and BIG5, have their dominant applica-

tions in different regions. GB is widely used in Mainland China and Singapore while BIG5 is the de facto standard in Hong Kong and Taiwan of China.

## GB

GB, which stands for “Guo Biao” (国标, National Standard), is actually the designator of a series of standards originated from the standard GB 2312-80, which was established on May 1, 1981 by the Chinese government with the official name “Code of Chinese Graphic Character Set for Information Interchange Primary Set”. GB 2312-80 contains 7,445 characters, including Chinese characters, full-width<sup>1</sup> ASCII characters, miscellaneous symbols, and others.

Several standards, GB 6345.1-86, GB8565.2, and ISO-IR-165:1992, were issued with corrections and extensions to GB 2312-80. Among them, ISO-IR-165:1992 is the CCITT (Comité Consultatif International Télégraphique et Téléphonique) Chinese set which enumerates 8,443 including all characters in the three GB standards. These standards are for simplified Chinese characters. There is also GB/T 12345-90 that is for traditional Chinese characters.

The Unicode version 1.1, ISO 10646-1:1993, has its Chinese translation, GB 13000.1-93, which includes a Chinese character subset known as GBK. GBK is composed of characters in GB 6345.1-86, 14,240 additional Chinese characters, 166 additional symbols, and part (some non-Chinese characters) of GB/T 12345-90.

Common encoding methods for GB character sets include ISO-2202-CN, EUC-CN, HZ, and GBK. Both ISO-2202-CN and EUC-CN (Extended Unix Code) are widely used in encoding GB 2312-80 character set. The most important difference between the two methods is that ISO-2202-CN is seven-bit encoding while EUC-CN is eight-bit encoding. HZ encoding is a simplistic yet effective seven-bit encoding for GB 2312-80. It is mainly used in exchanging emails and Usenet news. GBK encoding is designated for encoding the GBK character set.

---

<sup>1</sup>“Full-width” means the characters occupy two bytes as all Chinese characters.

## Big5

Big Five character set was established on May 1, 1984 by the Institute for Information Industry of Taiwan. It enumerates 5,401 and 7,652 (13,053 in total) Chinese characters in two levels. It is not the national standard but the de facto standard used in Taiwan. Its encoding method is also called Big Five.

Correcting and extending Big Five, CNS 11643-1992 (CNS stands for Chinese National Standard) is the national standard issued in Taiwan. Enumerating 48,027 Chinese characters, it is by far the largest Chinese character set standard. Encoding methods used for CNS include ISO-2022-CN and EUC-TW. Even though CNS 11643-1992 is much larger and more correct than Big Five, Big Five is more widely used and it is the reason for many to continue to support it.

Both GB and Big Five are used in Hong Kong. Big Five is more popular and standardized in Hong Kong. Because there are some locally used characters that Big Five doesn't include, some companies extended Big Five. So did the Hong Kong government, which supplemented 3049 more Chinese characters by publishing the Hong Kong GCCS (Government Chinese Character Set).

## Unicode

For the convenience of exchanging text internationally, great effort has been made in the development of international character set standards that cover most of world's languages in a single set. Unicode is such a standard developed by the Unicode Consortium and ISO trying to provide "a consistent way of encoding multilingual plain text and brings order to a chaotic state of affairs that has made it difficult to exchange text files internationally" [Uni99]. Unicode is compatible with ISO 10646-1:1993. In fact, Unicode is equivalent to the BMP (Basic Multilingual Plane), the only non-empty plane of ISO 10646.

The latest version of the Unicode standard, Version 2.1, contains 38,887 characters from the principal languages all over the world. From the Chinese point of view, an important benefit of Unicode is it unified the Chinese characters from many CJK

character set standards into a single set.

There are a series of encoding methods for Unicode, including UCS-2, UCS-4, UTF-7, UTF-8, and UTF-16. Details of these encoding methods are beyond the scope of this thesis. The book, *CJKV information processing* [Lun99], is a very complete and well written reference. It also provides references to other useful resources.

### 3.1.2 Related techniques

Chinese processing is involved in all the stages in this work. In the parallel text mining, we adopted SILC [IFP97], a language and character set detection program, to identify language and character set of mined Web pages for the purpose of filtering and future processing (see the last chapter). In the translation model training, we have to unify the Chinese texts into a single coding, GB in this case. Chinese segmentation is also a very important step, which we will discuss later in this chapter.

Code conversion algorithm varies with different coding pairs. In our case, what we have are mostly Big5 texts and some GB texts. Because the segmentation dictionary is in GB and so is the Chinese collection for the future trans-lingual IR experiments, we converted the Big5 texts into GB code. Strictly, it is impossible to convert Big5 code to GB because the Big5 character set has a lot more characters than GB 2312. However, since most characters we encounter are in both sets, the conversion is feasible. The code conversion tool we used is NCF (Network Hanzi Filter) <sup>2</sup>.

Here we would particularly emphasize Java's power of processing multi-character-set information. A great feature of Java is that it uses Unicode internally. Before being processed, input encoded by numerous methods are transferred to Unicode by the standard Java I/O, and vice versa. One can practically write any code converter between the encoding methods Java supports in a few minutes.

As an example, we show a small utility, *bd2punc*, which converts full-width (two bytes) Chinese punctuation to punctuation in ASCII. As we will see later, before text alignment, the texts have to be divided into separate sentences. We do so

---

<sup>2</sup><ftp://ftp.net.tsinghua.edu.cn/pub/Chinese/ncf/>.

partially according to punctuation. However, the punctuation in Chinese text are normally full-width and the available Perl script for sentencing only recognizes single-byte characters. An easy way to get around this problem is to convert full-width punctuation to its corresponding single-byte one. This is where *bd2punc*<sup>3</sup> comes in.

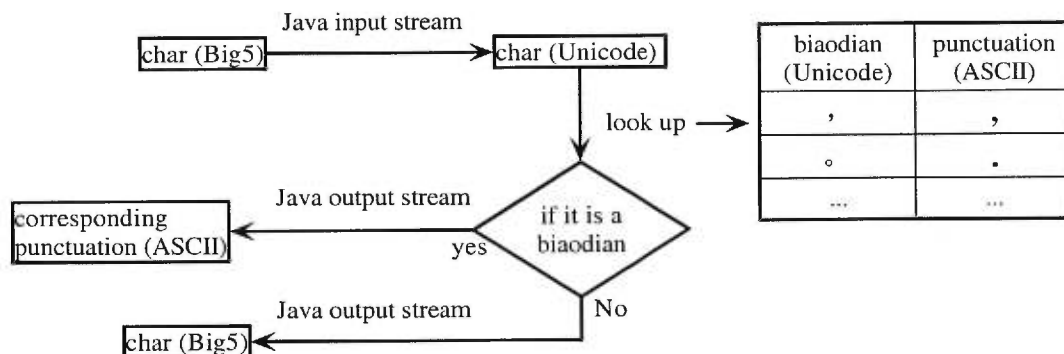


Figure 3.3: Convert biaodian (Chinese punctuations) to its corresponding ASCII punctuation.

Fig. 3.3 illustrates how *bd2punc* works. It reads Chinese text (in Big5 or GB) through Java input stream which converts the text to Unicode. The following code shows the definitions of I/O streams for specific character set.

```
File inFile = new File(inputFile);
FileInputStream inStream = new FileInputStream(inFile);
BufferedReader in = new BufferedReader(
    new InputStreamReader(inStream, charSet));

File outFile = new File(outputFile);
FileOutputStream outputStream = new FileOutputStream(outFile);
BufferedWriter out = new BufferedWriter(
    new OutputStreamWriter(outputStream, charSet));
```

To convert biaodian to punctuation, we establish a map between them as showed in the table of Fig. 3.3. The program simply compares every input character to the biaodians in the table and then decides what to output. What's worthy of mentioning is how the punctuation table is established. There are actually two tables, one for GB and one for Big5. Each table contains objects of class *punc\_bd* defined in the

<sup>3</sup>“bd” stands for BiaoDian, meaning punctuation in Chinese.

following code, where the function *CodeOfBiaodian* shows how non-Unicode bytes are converted to Unicode, which is useful in processing texts in various codes.

```
public class punc_bd extends java.lang.Object
{
    private char punctuation;
    private byte[] biaodian;
    private String biaodianCharSet;
    private int unicodeOfBiaodian;

    private int CodeOfBiaodian(byte[] biaodian, String charSet)
        throws Exception
    {
        String biaodian_unicode = new String(biaodian, charSet);
        return (int)biaodian_unicode.charAt(0);
    }

    public punc_bd(String biaodian, char punctuation, String charSet)
        throws Exception
    {
        byte[] bytes = {(byte)biaodian.charAt(0),
                        (byte)biaodian.charAt(1)};
        this.biaodian = bytes;
        this.punctuation = punctuation;
        this.biaodianCharSet = charSet;
        this.unicodeOfBiaodian = CodeOfBiaodian(this.biaodian,
                                                biaodianCharSet);
    }

    public char punc() { return punctuation; }
    public int biaodian() { return unicodeOfBiaodian; }
}
```

## 3.2 Aligning the Parallel Corpus

Aligning the parallel corpus is a critical step of the pre-training process. Between each pair of texts, we generate a sentence level mapping. The precision of the alignment is vital to the performance of the translation model. Text alignment has been a very interesting and challenging topic in the last decade. We will first review briefly the important text alignment algorithms proposed in the past years. Then we will further discuss the method (by Simard et al.) we adopted as well as its interesting effect in

aligning English-Chinese web pages.

### 3.2.1 Introduction to Text Alignment Methods

Parallel text or bitext alignment is of great interest for people who intend to exploit translation information from existing translations between various languages. Its objective is to find a mapping between the units of two texts that are translations of each other. In terms of the size of text unit, there are paragraph level, sentence level, word or even character level alignment. In the word level, the alignment could be not exhaustive, i.e., not every word is mapped to a word in another language. In practice, the aligned text units are usually sentences. This is a good compromise between refinement in text units and the difficulty to do so.

Even at sentence level, bitext alignment represent several challenges:

- 1 A sentence is not always translated into one single sentence in another language. One sentence could correspond to several sentences. It is also quite frequent that relationships are many-many.
- 2 The texts we are dealing with are noisy. There could be omissions (some sentences are not translated) or additions (some sentences are added during translation without source text).

Several alignment algorithm have been developed, using length or/and cognates as criteria.

#### Pure Length-Based Methods

Great efforts have been done in text alignment throughout the last decade. The methods proposed can be classified into length-based statistical methods, lexical methods, or hybrids of the last two kinds. Two well-known statistical methods are presented by Brown et al. [BLM91] and Gale & Church [GC91]. Both methods try to find the most possible alignment using only sentence length and dynamic programming . The idea simply comes from the observation that source sentence(s) and its(their) translations

tend to have similar lengths and usually appear in the same or similar order. The main difference between the two methods is that Brown et al. count sentence length by words while Gale & Church use numbers of characters.

### Methods Using Cognates

Both of the two above methods were proved to be successful in aligning the Canadian Hansard corpus which is rather clean and easy to align. However, as pointed out by Simard et al. [SFI92] and Chen [Che93], while aligning more noisy corpora, the methods based solely on sentence length are not robust enough to cope with the above-mentioned difficulties. Simard et al. proposed a method that uses lexical information, cognates, to help with alignment [SFI92].

“Cognates are pairs of tokens of different languages which share obvious phonological or orthographic and semantic properties, with the result that they are likely to be used as mutual translations.” [SFI92] Examples are *generation/génération* and *financed/financié* for English/French. In a wider sense, cognates can also include numerical expressions and punctuation.

Instead of defining a specific list of cognates for each language pair, Simard et al. gave language-independent definitions on cognates. Cognates are recognized on the fly according to a series of rules. For example, words starting with 4 identical letters in English and French are considered as cognates. They also proposed a way to numerically measure “cognateness”, i.e., how two segments of text are related in terms of cognates. Experiments showed that aligning only based on cognateness level is not as accurate as length-based method. Better results were obtained when cognates were combined with length. In fact, the method consists of two passes. The first pass uses pure length criterion to generate a list of best scored alignments. The second pass then uses cognateness score function to select the best alignment from the list.

For the same purpose of “balancing robustness and accuracy”, Simard et al. also suggested an opposite way of combining cognate-based and length-based methods



[SP98]. This time, a cognate-based program is applied first to provide a “reliable and robust” character-level mapping. The resulted mapping is then passed to a length-based program as the search space. Based on the search space, the length-based program is supposed to give “accurate” sentence-level alignments.

Inspired by the cognate-based approach of Simard et al., Church developed the program *Char\_align*, a character level alignment program using cognates [Chu93]. This program does not require a priori sentence segmentation which might be wrong in noisy corpora such as OCR output and lead to false alignment.

### Methods Using Words as Lexical Information

In addition to using cognates as lexical information, another option is to use words. For example, Chen’s algorithm establishes a statistical word translation model and search for the alignment that “maximizes the probability of generating the corpus with the model” [Che93]. Kay and Röscheisen developed an algorithm which iteratively generates word level and sentence level alignments [KR93]. Each iteration helps to refine the results in the next iteration. The results were found to converge to the correct alignment.

Finding word correspondences for non-alphabetic languages such as Chinese is especially challenging. Obviously, there are no cognates to find in Chinese to help with alignment. Although punctuation can be taken as cognate, it is not of much help alone. Wu experimented Gale and Church’s length-based method on Hong Kong Hansard and obtained much poorer results than that of Canadian Hansard [Wu94]. This result is predictable because of the great difference between East Asian and European languages.

To improve the alignment performance on the Chinese-English corpus, Wu incorporated a small lexicon as “lexical cues” into the length-based method [Wu94]. Significant improvement was achieved in part of the corpus where most errors occurred with the length-based method. The lexicon used was manually constructed.

Fung and McKeown introduced DK-vec, an algorithm for producing a small bilin-

gual lexicon from parallel texts according to “frequency, position and recency information” [FM94]. Similar to *Char-align*, the DK-vec algorithm does not consider natural sentence boundaries so as to avoid errors caused by noises in corpus. The generated corpus could be used as anchor points in the next pass of alignment.

### 3.2.2 Using Markups as Cognates

The alignment task in this project is a challenging one. What we are facing is a highly noisy English-Chinese hypertext corpus. We encounter the following difficulties.

- The corpus contains mostly parallel pages and a small portion of non-parallel ones that cannot be aligned at all. Web pages are not like formal documentation. Their translation quality varies from site to site, or even from page to page. There are great chances that deletions or additions exist.
- The great difference between the syntactic structures and writing systems of English and Chinese makes the situation even worse. As mentioned above, previous experiments applying methods designed for European languages on Asian languages such as Chinese showed poorer results.

Fortunately, the objective of the alignment is to provide training material to the statistical translation model which can tolerate noise to some extent. Incorrect alignments will not affect the model seriously as long as there are sufficient correct ones. The recall ratio of the alignment is as important as its precision. Thus what we need is a robust method that can cope with noises and keep a reasonable overall precision/recall ratio. In other words, when an alignment error occurs in some location, instead of letting it lead to errors on all other alignments, the method should be able to limit its effect in its smallest vicinity. In the methods introduced above, cognates or word mapping are used to achieve this goal. They may prevent the spreading or errors to subsequent sentences. An example will be given later to show this.

The method we adopted in this work was proposed by Simard et al. [SFI92] which takes both length similarity and cognateness as its criteria. We described the basic

idea of the method in the previous subsection. For details of the alignment algorithm, see [SFI92]. Here we would focus on its application in Chinese-English HTML text alignment.

As we are aware, Chinese is not an alphabetic language and thus it's impossible to extract cognates from Chinese characters. However, what we are dealing with are hypertexts, i.e., texts with markups from which we can exploit alignment information. Cognates can be extracted from the HTML markups in both English and Chinese pages of a parallel pair. Since they most probably have similar HTML structures, the cognates can help with alignment.

To illustrate how markups help with the alignment, we align the same pair with both the pure length-based method of Gale & Church (Fig. 3.4, Fig. 3.5), and the method of Simard et al. (Fig. 3.6, Fig. 3.7). First of all, we observe from the figures that the two texts are divided into sentences. The sentences are marked by `<s id="xxxx">` and `</s>`. Note that we determine sentences not only by periods, but also HTML markups.

In this example, we further notice that this pair can be easily aligned except for sentence 0002 whose Chinese version is much longer than its counterpart in the English page. The length-based method thus take sentence 0002, 0003, and 0004 in the English page as the translation of sentence 0002 in the Chinese page (Fig. 3.4), which is wrong. It also caused the three following incorrect alignments. As we can see in Fig. 3.6, the cognate method did not make mistake because of the noise at sentence 0002. Despite their large length difference, the two 0002 sentences are still aligned as a 1-1 pair, because the sentences in the following 4 alignments (0003 - 0003; 0004 - 0004, 0005; 0005 - 0006; 0006 - 0007) have rather similar HTML markups and are believed by the program to be very likely the correct alignments.

The cognate method of Simard et al. was developed for aligning alphabetic languages, especially languages that share a lot cognates such as English and French. It is not suitable for aligning natural English-Chinese bitext. It is the HTML markups that make it possible to obtain acceptable results with this method. It would be very interesting to compare the results with what Wu obtained in his experiment [Wu94],

in which a small lexicon was used. However, because the two experiments are done on different corpora, a direct comparison between the two alignment methods is impossible now.

<pre>&lt;s id="0000"&gt; &lt;HTML&gt; &lt;HEAD&gt; &lt;META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=iso-8859-1"&gt; &lt;META HTTP-EQUIV="Content-language" CONTENT="Western"&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0000"&gt; &lt;HTML&gt; &lt;HEAD&gt; &lt;META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=big5"&gt; &lt;META HTTP-EQUIV="Content-language" CONTENT="zh"&gt; &lt;/s&gt;</pre>
<pre>&lt;s id="0001"&gt; &lt;TITLE&gt;Journal of Primary Education 1996, Vol., No. 1&amp;2, pp. 19-27 &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0001"&gt; &lt;TITLE&gt; Journal of Primary Education 1996, Vol., No. 1&amp;2, Page 19-27 &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;/s&gt;</pre>
<pre>&lt;s id="0002"&gt; &lt;BODY BACKGROUND="../gif/pejbg.jpg" TEXT="#000000" BGCOLOR="#ffffff"&gt; &lt;CENTER&gt; &lt;/s&gt; &lt;s id="0003"&gt; &lt;H1&gt;Journal of Primary Education &lt;/H1&gt; &lt;/s&gt; &lt;s id="0004"&gt; &lt;HR&gt; &lt;B&gt;Volume 6, No 1&amp;2, pp. 19-27 (May, 1996) &lt;/B&gt; &lt;HR&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0002"&gt; &lt;BODY BACKGROUND="../gif/pejbg.jpg" TEXT="#000000" BGCOLOR="#ffffff"&gt; &lt;A HREF="/en/pej/b2g_pej.phtml?URL=%2fen%2fpej%2f0601%2f0601019c.htm"&gt; &lt;IMG SRC="/en/gif/kan.gif" ALT="简体" BORDER=0 ALIGN=R IGH&gt; &lt;/A&gt; &lt;CENTER&gt; &lt;/s&gt;</pre>
<pre>&lt;s id="0005"&gt; &lt;H3&gt;Principles for Redesigning Teacher Education &lt;/H3&gt; Alan TOM &lt;/CENTER&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0003"&gt; &lt;H2&gt;初等教育学报&lt;/H2&gt; &lt;/s&gt; &lt;s id="0004"&gt; &lt;HR&gt; (一九九六年五月) 第六卷. &lt;/s&gt;</pre>
<pre>&lt;s id="0006"&gt; &lt;P&gt; &lt;B&gt;&lt;I&gt; Abstract &lt;/I&gt; &lt;/B&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0005"&gt; 第一及二期19-27页 &lt;HR&gt; &lt;/s&gt;</pre>

Figure 3.4: An alignment example using pure length-based method.

<p>&lt;s id="0007"&gt; &lt;BLOCKQUOTE&gt; In order to establish a basis for redesigning initial teacher preparation, the author examines two instances during his career when he was excited in his work as a teacher educator. &lt;/s&gt;</p>	<p>&lt;s id="0006"&gt; &lt;H3&gt;革新教师教育的原则 &lt;/H3&gt; Alan TOM &lt;/CENTER&gt; &lt;/s&gt; &lt;s id="0007"&gt; &lt;P&gt; &lt;I&gt; &lt;B&gt; 摘要 &lt;/B&gt; &lt;/I&gt; &lt;P&gt; &lt;/s&gt; &lt;s id="0008"&gt; &lt;BLOCKQUOTE&gt; 作者检视他过往热心地训练准教师时的两个事例，作为改革教师教育的根据。 &lt;/s&gt;</p>
<p>&lt;s id="0008"&gt; He also analyzes external barriers to achieving excitement in teacher education, including not only state and national regulation of teacher education but also the low status of teacher education in the United States. &lt;/s&gt;</p>	<p>&lt;s id="0009"&gt; 作者分析导致美国的教师教育了无生气的外在原因，除了州政府和国家颁下的规定外，还有教师社会地位的低微。 &lt;/s&gt;</p>
<p>&lt;s id="0009"&gt; Barriers are also created by three questionable beliefs which are widely accepted among teacher educators, namely, that subject matter and pedagogy are separable, that pedagogical knowledge is critically important in teacher education, and that specialized knowledge is the core of pedagogical knowledge. &lt;/s&gt;</p>	<p>&lt;s id="0010"&gt; 作者认为教师教育存在三个不正确的想法，即是，内容与教学法是分割的；教学知识至为重要；教学知识的核心为专门知识。 &lt;/s&gt; &lt;s id="0011"&gt; 这些想法更成为教师教育的障碍。 &lt;/s&gt;</p>
<p>&lt;s id="0010"&gt; Eleven principles for redesigning teacher education are proposed as a way both of capturing the essential attributes of exciting teacher education and of addressing the external barriers and the three mistaken beliefs. &lt;/s&gt; &lt;s id="0011"&gt; Also considered are the reform ideas of John Goodlad and the attempt to encourage reform by posing the question of what teacher education is fruitful is yet to be determined; additional experience with reform efforts is needed. &lt;/s&gt;</p>	<p>&lt;s id="0012"&gt; 有鉴于此，作者建议了十一项革新教师教育的原则。 &lt;/s&gt; &lt;s id="0013"&gt; 同时，作者亦提倡 John Goodlad 的改革意见，以及有关教师的应有知识与技能的改革的尝试。 &lt;/s&gt; &lt;s id="0014"&gt; 他以为改革能否成功似是言之尚早，因为我们还需更多有关改革的经验和知识。 &lt;/s&gt;</p>
<p>*** **</p>	<p>*** **</p>

Figure 3.5: An alignment example using pure length-based method (continued).

<pre>&lt;s id="0000"&gt; &lt;HTML&gt; &lt;HEAD&gt; &lt;META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=iso-8859-1"&gt; &lt;META HTTP-EQUIV="Content-language" CONTENT="Western"&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0000"&gt; &lt;HTML&gt; &lt;HEAD&gt; &lt;META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=big5"&gt; &lt;META HTTP-EQUIV="Content-language" CONTENT="zh"&gt; &lt;/s&gt;</pre>
<pre>&lt;s id="0001"&gt; &lt;TITLE&gt;Journal of Primary Education 1996, Vol., No. 1&amp;2, pp. 19-27 &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0001"&gt; &lt;TITLE&gt; Journal of Primary Education 1996, Vol., No. 1&amp;2, Page 19-27 &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;/s&gt;</pre>
<pre>&lt;s id="0002"&gt; &lt;BODY BACKGROUND=" ../gif/pejbg.jpg" TEXT="#000000" BGCOLOR="#ffffff"&gt; &lt;CENTER&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0002"&gt; &lt;BODY BACKGROUND=" ../gif/pejbg.jpg" TEXT="#000000" BGCOLOR="#ffffff"&gt; &lt;A HREF="/en/pej/b2g_pej.phtml?URL=%2fen%2fpej%2f0601%2f0601019c.htm"&gt; &lt;IMG SRC="/en/gif/kan.gif" ALT="简体" BORDER=0 ALIGN=R IGH&gt; &lt;/A&gt; &lt;CENTER&gt; &lt;/s&gt;</pre>
<pre>&lt;s id="0003"&gt; &lt;H1&gt;Journal of Primary Education &lt;/H1&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0003"&gt; &lt;H2&gt;初等教育学报&lt;/H2&gt; &lt;/s&gt;</pre>
<pre>&lt;s id="0004"&gt; &lt;HR&gt; &lt;B&gt;Volume 6, No 1&amp;2, pp. 19-27 (May, 1996) &lt;/B&gt; &lt;HR&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0004"&gt; &lt;HR&gt; (一九九六年五月) 第六卷. &lt;/s&gt; &lt;s id="0005"&gt; 第一及二期19-27页 &lt;HR&gt; &lt;/s&gt;</pre>
<pre>&lt;s id="0005"&gt; &lt;H3&gt;Principles for Redesigning Teacher Education &lt;/H3&gt; Alan TOM &lt;/CENTER&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0006"&gt; &lt;H3&gt;革新教师教育的原则 &lt;/H3&gt; Alan TOM &lt;/CENTER&gt; &lt;/s&gt;</pre>
<pre>&lt;s id="0006"&gt; &lt;P&gt; &lt;B&gt; &lt;I&gt; Abstract &lt;/I&gt; &lt;/B&gt; &lt;/s&gt;</pre>	<pre>&lt;s id="0007"&gt; &lt;P&gt; &lt;I&gt; &lt;B&gt; 摘要 &lt;/B&gt; &lt;/I&gt; &lt;P&gt; &lt;/s&gt;</pre>

Figure 3.6: An alignment example considering cognates.

<p>&lt;s id="0007"&gt; &lt;BLOCKQUOTE&gt; In order to establish a basis for redesigning initial teacher preparation, the author examines two instances during his career when he was excited in his work as a teacher educator. &lt;/s&gt;</p>	<p>&lt;s id="0008"&gt; &lt;BLOCKQUOTE&gt; 作者检视他过往热心地训练准教师时的两个事例, 作为改革教师教育的根据. &lt;/s&gt;</p>
<p>&lt;s id="0008"&gt; He also analyzes external barriers to achieving excitement in teacher education, including not only state and national regulation of teacher education but also the low status of teacher education in the United States. &lt;/s&gt;</p>	<p>&lt;s id="0009"&gt; 作者分析导致美国的教师教育了无生气的外在原因, 除了州政府和国家颁下的规定外, 还有教师社会地位的低微. &lt;/s&gt;</p>
<p>&lt;s id="0009"&gt; Barriers are also created by three questionable beliefs which are widely accepted among teacher educators, namely, that subject matter and pedagogy are separable, that pedagogical knowledge is critically important in teacher education, and that specialized knowledge is the core of pedagogical knowledge. &lt;/s&gt;</p>	<p>&lt;s id="0010"&gt; 作者认为教师教育存在三个不正确的想法, 即是, 内容与教学法是分割的; 教学知识至为重要; 教学知识的核心为专门知识. &lt;/s&gt;</p>
<p>&lt;s id="0010"&gt; Eleven principles for redesigning teacher education are proposed as a way both of capturing the essential attributes of exciting teacher education and of addressing the external barriers and the three mistaken beliefs. &lt;/s&gt;</p>	<p>&lt;s id="0011"&gt; 这些想法更成为教师教育的障碍. &lt;/s&gt; &lt;s id="0012"&gt; 有鉴于此, 作者建议了十一项革新教师教育的原则. &lt;/s&gt;</p>
<p>&lt;s id="0011"&gt; Also considered are the reform ideas of John Goodlad and the attempt to encourage reform by posing the question of what teacher education is fruitful is yet to be determined; additional experience with reform efforts is needed. &lt;/s&gt;</p>	<p>&lt;s id="0013"&gt; 同时, 作者亦提倡 John Goodlad 的改革意见, 以及有关教师的应有知识与技能的改革的尝试. &lt;/s&gt; &lt;s id="0014"&gt; 他以为改革能否成功似是言之尚早, 因为我们还需更多有关改革的经验和知识. &lt;/s&gt;</p>
<p>... ..</p>	<p>... ..</p>

Figure 3.7: An alignment example considering cognates (continued).

We believe there is room to improve the alignment of the corpus. The definition of cognate in the current method could be improved to make better use of the markups. It is also worthy of experimenting to incorporate a lexicon into the current method, i.e., consider also word correspondence from a manual lexicon as in Wu et al. [Wu94].

### 3.3 Chinese Segmentation and English Expression Extraction

The results of text alignment are two sets of sentences and the mapping data between them. Before feeding the data to the translation model, some processing on the raw sentences are necessary. For example, the English words have to be transferred to their citation forms (singular form for nouns, infinitive form for verbs). Besides, there are two important processing: Chinese segmentation and English expression extraction.

#### 3.3.1 Chinese Segmentation

One of the many difficulties in processing Chinese texts is that a Chinese sentence is a string of Chinese characters without spaces between them to indicate word boundaries. In fact, the situation is even worse in ancient Chinese texts, which don't even have punctuation. Sentence boundaries were up to the reader to decide. In tasks such as machine translation, information retrieval, or information extraction, word instead of character is assumed to be the basic analysis unit of text. Thus segmentation of sentences into words has to be done before any further processing.

Liu claimed that the two main difficulties of Chinese segmentation are the vagueness of word definition and the “word chain” problem [Liy87]. Words are very weakly defined in Chinese. There cannot be a complete dictionary that contains all the words. Words can be domain dependent, and new words are being invented constantly. A word chain is a sequence of Chinese characters that could produce overlapping words. It causes ambiguities in segmentation.



In the past decades, many Chinese segmentation approaches were studied. Two main categories are the dictionary-based approaches [LZ91, CK92] and the statistical approaches [Ca91, SS91]. Dictionary-based approaches rely on dictionaries that cover the most usual words and heuristic rules that correspond to common word structures. Even though heuristic rules can find some compound words that are not included, the dictionary used still has to be rather complete to guarantee high-quality segmentation results. The statistical approaches, on the contrary, do not require dictionaries. They learn statistical information such as word occurrence frequencies from manually segmented corpora. The coverage and accuracy of the training corpora are then crucial to the performance of segmentation. Some hybrid approaches combining the last two methods were also suggested. For example, Nie et al. proposed an approach which flexibly incorporates statistical information (if available) with dictionaries and heuristic rules [NJH94, NRB95].

In this work we use a dictionary-based segmentation program. The program uses two large dictionaries including 87,600 and 187,182 words (words in the two dictionaries may overlap). It also incorporates a set of heuristic rules to find words in common structures such as nominal pre-determiner structure and affix structure. The maximum-matching algorithm is used. From all the possible segmentation choices, the algorithm simply selects the one with the fewest words. This simple algorithm surely cannot totally solve the ambiguity problem. More sophisticated disambiguation techniques could be used [Jin92, Jin94]. However, for IR, a simple segmentation process will suffice. A particularity of our segmentation is that we could extract not only the long words, but also the short words included in the long words. For example, if “ABCD”, “AB” and “CD” are all words, they will all be extracted. This is done to allow higher recall ratio for IR.

### 3.3.2 English Expression Extraction

English words are explicitly separated by spaces. English texts do not need to be segmented. On the contrary, some sequences of words (expressions) (e.g., “in order

to”), have to be extracted and fed to the translation model as a single entry so as to enable the translation model to recognize them.

We extract expressions by comparing the input words with the available dictionary of expressions. All the matching expressions are reformatted into a single word by replacing spaces with ‘\_’. For example, “catch up” is transferred to “catch\_up”. Making use of hash table, a small program can process large amount of text in a short time.

To make sure that all the expressions are found, the English words in both the texts and the dictionary are transferred to lower-case characters and their citation forms. As the result, the expression “United States of America” is in the form “united state of america” in the dictionary. Besides, before extraction we also filter out the HTML markups (they are not useful any more); illegal words such as mixture of characters, numbers and symbols; and useless punctuation. Abbreviations such as “u.s.a.” and compound words such as “multi-disciplinary” are preserved. We remove the “s”s in some words unless the word constitutes an expression with the following word. For example, we replace “father’s” by “father”. Most of the time it is the word “father” that we want to input to the translation model except when the next word is “day”. In that case, we have the expression “father’s\_day”.

Except the prepositions, the words constituting expressions have their own meanings. To ensure the translation coverage on these words, one option is to provide both expressions and the words constituting them to the translation model. That is, from “father’s day”, we recognize not only “father’s\_day”, but also “father” and “day”.

### 3.4 Translation Model Training

After the pre-training processes, final training source in the form as shown in Fig. 3.2 are learnt by the statistical translation model. We will briefly describe the principles of the model. Even though the intended application of the model is translating queries in CLIR, we first evaluate and fine-tune the model by generating evaluation lexicons and observe their precision.

### 3.4.1 Principles of the Statistical Model

Learning from the previous translation examples by human is often used in machine translation. Most work of this kind establishes probabilistic models from parallel corpora. Based on one of the statistical models proposed by Brown et al. [BPPM93], the basic principle of our translation model is: given aligned translations, if two words often co-occur in the source and target sentences, there is a high chance that they are translations of each other. Specifically, the model learns (from a large set of alignments) the probability,  $p(t|s)$ , of having a word  $t$  in the translation of a sentence containing a word  $s$ . For an input sentence, the model then calculates a sequence of words that are most probable in its translation.

We briefly describe the model training as follows. For a single alignment  $a_k$  between the source sentence  $S$  and the target sentence  $T$ , we have two sets of words:

$$S = \{s_1, s_2, s_3, \dots, s_l\},$$

$$T = \{t_1, t_2, t_3, \dots, t_m\}.$$

We regard each word  $t_j$  in  $T$  as probable translation of each word  $s_i$  in  $S$ . All the possibilities are treated as equivalent. We then have

$$p(t_j|s_i, a_k) = C_T/l$$

where  $C_T$  is a parameter related to the length of the target sentence. Now, for a set of alignments  $A$ , we calculate the overall probability  $p(t_j|s_i, A)$  from all  $p(t_j|s_i, a_k)$  by

$$p(t_j|s_i, A) = C_A \sum_k p(t_j|s_i, a_k)$$

where  $C_A$  is a normalization factor. With the Expectation Maximization algorithm, the probability  $p(t_j|s_i)$  is finally determined from  $p(t_j|s_i, A)$ .

Given a sentence  $S$ , the probability of having word  $t$  in its translation is determined by all the words in  $S$ . In fact,

$$p(t|S) = C_S \sum_i p(t|s_i)$$

where  $C_s$  is another normalization parameter related to the length of  $S$ . For a sentence, the model provides a series of most probable translations, i.e., words with the highest probabilities of being the sentence’s translation.

As mentioned above, for an alignment, every word in the target sentence is considered to be equivalently the possible translation of any word in the source sentence. The training process does not take the word positions into account. This assumption implies that the translation model does not learn syntactic information from the training source and thus cannot be used to obtain syntactically correct translations. However, the model fits the need of cross-language information retrieval which does not require translated queries to be syntactically correct, but only to find out the most important translation words.

### 3.4.2 Analysis on Evaluation Lexicons

The training corpus produced by PTMiner system includes 117.2M Chinese text and 136.5M English text. Tab. 3.1 lists the statistics of the training source (after the pre-training process) for the testing model. There are 1,048,156 sentence alignments in total and 870,414 of them are 1-1 alignments that are believed to be more likely accurate. The testing models mentioned in this section are all trained with only the 1-1 alignments.

	size	vocabulary	word count
src.e	74.1M	76,969	9,816,859
src.c	51.1M	48,528	9,916,416

Table 3.1: Statistics of the training source.

Note that both the English and Chinese vocabularies contain illegal words. For example, the spelling errors in English words and the telephone numbers (in full-width characters) in Chinese words. Even though the illegal words are often seen in the vocabularies, each single one of them doesn’t appear frequently in the training source. There is only a very small portion of the word counts and will not affect the

performance of the translation model. In fact, words with very low frequency are not even taken into account by the model. In the implementation of model training, a frequency threshold may be set to filter out rare words.

The ultimate goal of establishing this translation model is to provide a context and domain sensitive translator for the CLIR queries. It is expected that a more precise model can be obtained if we consider groups of words. However, we can still investigate the performance of the model in CLIR that only consider single words. Some similar work has been done to automatically extract lexicons from parallel corpora [WX94, Wu95, Fun, Bro98]. For example, using similar statistical model, Wu extracted an English-Chinese lexicon with encouraging precision from the Hong Kong Hansard which has better parallel quality than our corpus. It is very interesting to see if we can learn lexicons from a noisy yet large corpus collected from the Web and to compare it with the work of Wu.

To measure the precision of the translation model, we randomly selected 200 Chinese words and 200 English words from the training source. Note that we picked these words from the training source instead of the vocabularies. As a result, the chosen words tend to be the words that frequently appear in the training source. We are more concerned about the precision of translating these words, which will also be frequently met in the CLIR queries. If we randomly chose from the vocabularies, we would obtain illegal words or low frequency words which won't (or very unlikely) have to be translated in the future.

For each word, the translation model gives a series of most probable translations as well as their probabilities. We evaluate the precision of the most probable translation of each word. A translation is considered as correct if

- 1 the most probable translation (the first one) is correct; or
- 2 the first couple of translations constitute a correct translation for the word if the word should be translated by a group of words.

The second condition is set because some words do not have exact correspondences in the other language in single words. For instance, “自由民主党” is a single word in

the Chinese dictionary while its English translation, “liberal democratic party”, has to be in three words. We assume the translation is right if the first three translations are “liberal”, “democratic”, and “party”, no matter what the order is.

With the criterion mentioned above, we found the performance of the translation model encouraging. Despite the noisy nature of the corpus, we get the precision of the most probable translation at 81.5% for the English-Chinese translations and 77% for the Chinese-English translations.

### Effects of Stop-lists

We also found that the incorporation of the stop-lists has a significant effect on the translation model.

A Stop-list is a set of the most frequent grammatical words that we remove from the training source. These words are not of interest for IR. Because these words exist in most alignments, the statistical model cannot conclude correct translations for them. More importantly, their existence greatly affects the accuracy of other translations. They can be taken as the translation for many words.

At first glance, it seems that both the English and Chinese stop-lists should be applied to eliminate the noises caused by them. Interestingly, we found that the effect of stop-lists depends on the translation direction. Under some condition we might want to keep the words in stop-list.

model	En. stoplist	Ch. stoplist	En. to Ch.	Ch. to En.
1	yes	yes	79.5%	75%
2	no	yes	81.5%	N/A
3	yes	no	N/A	77%
4	no	no	63%	72.5%

Table 3.2: Precision of testing models.

words	model1 (with both stoplists)		model2 (with Chinese stoplist)		model4 (with no stoplists)	
a.m.	t	上午	t	上午	f	的
access	f	公开	f	公开	f	的
adaptation	t	适应	t	适应	t	适应
add	t	补充	t	补充	t	补充
adopt	t	采用	t	采用	t	采用
agent	t	代理人	t	代理人	t	代理人
agree	t	同意	t	同意	t	同意
airline	t	航空公司	t	航空公司	t	航空公司
amendment	t	修订	t	修订	t	修订
appliance	t	用具	t	用具	f	的
apply	t	适用	t	适用	t	适用
attendance	t	列席	t	列席	t	列席
auditor	f	审核	f	审核	f	的
average	t	平均	t	平均	t	平均
base_on	f	计算	f	计算	t	根据
block	f	大厦	f	大厦	f	的
bottom	t	最低	t	最低	f	的
break_law	f	候选	f	冒险	f	的
breath	t	呼气	t	呼气	t	呼气
briefing	t	简报	t	简报	t	简报
building	t	建筑物	t	建筑物	f	的
business	t	业务	t	业务	f	的
carry	f	政府	f	工程	t	进行
category	t	类别	t	类别	f	的
census	f	政府	t	统计	f	的

Figure 3.8: E-C translations.

words	model1 (with both stoplists)		model3 (with English stoplist)		model4 (with no stoplists)	
办事处	t	office	t	office	t	office
保护	t	protection	t	protection	t	protection
报告	t	report	t	report	t	report
备	t	prepare	t	prepare	t	prepare
本地	t	local	t	local	t	local
便会	f	follow	f	follow	t	will
标准	t	standard	t	standard	t	standard
补校	f	adult	f	adult	f	of
不足	t	inadequate	t	inadequate	f	of
部分	t	part	t	part	t	some
财经	t	financial	t	financial	t	financial
参观	t	visit	t	visit	t	visit
草案	t	bill	t	bill	t	bill
车辆	t	vehicle	t	vehicle	t	vehicle
储蓄	t	saving	t	saving	t	saving
处理	t	deal	t	handle	t	with
传真	t	fax	t	fax	t	fax
次序	t	order	t	order	t	order
措施	t	measure	t	measure	t	measure
达到	t	achieve	t	achieve	t	achieve
当局	t	administration	t	administration	t	administration
登记	t	registration	t	registration	t	registration
电子	t	electronic	t	electronic	t	electronic
调	t	adjust	t	adjust	f	of
定	t	determine	t	determine	f	of

Figure 3.9: C-E translations.



Tab. 3.2 lists the precision we obtained from 4 different models using or not using stop-lists. We can see that for both lexicons, the highest precision is not obtained from the model using both stop-lists. Taking the English-Chinese lexicon as an example, among the three models the model using no stop-lists has the lowest precision. The best model is the one using the Chinese stop-list but not the English one. The Chinese-English lexicon has the symmetric phenomenon. The model using only the stop-list of the target language has the highest precision. Note that we didn't calculate the precision of applying model 2 to C-E translation and model 3 to E-C translation because those two cases have too poor precision, as will be proved in Fig. 3.12. They are the two worst cases for the E-C and C-E translation.

Fig. 3.8 (English-Chinese) and Fig. 3.9 (Chinese-English) show the first 25 translations as well as their probabilities given by different models. The 't's and 'f's indicate if they are true or false translations. From Fig. 3.8 we observe the following characteristics:

- 1 In model 4, which uses no stop-lists, words such as “of”, “with”, and “some” are mistaken as the translations for many words.
- 2 For most words which the three models give the same translation, the translation given by model 2 has the highest probability while the one given by model 4 has the lowest.

Similar phenomenon can be observed from Fig. 3.9 for the Chinese-English lexicon.

The first phenomenon can be possibly explained as follows. In Fig. 3.10, if the Chinese word  $C$  exists in the same alignments with the English word  $E$  more than any other Chinese words,  $C$  will be the most probable translation for  $E$ . Because of their frequent appearance, some words in the Chinese stop-list may have many or more chances to be in the same alignments with  $E$ . The probability of the translation  $E \rightarrow C$  is then reduced (maybe less than those of the incorrect ones). That is why words like “of” or “with” become the most probable translations for many words in model 4 of Fig. 3.8.

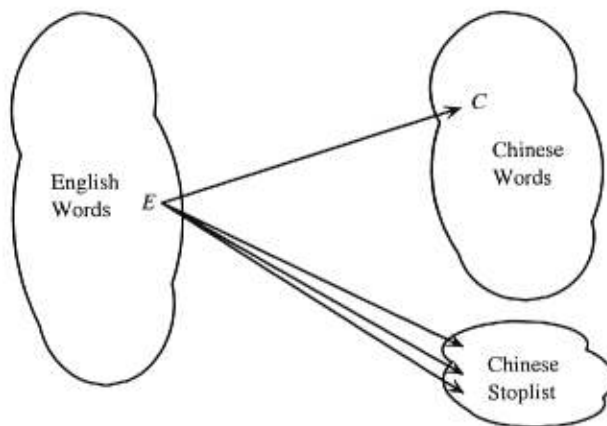


Figure 3.10: Effect of English stop list in C-E translation.

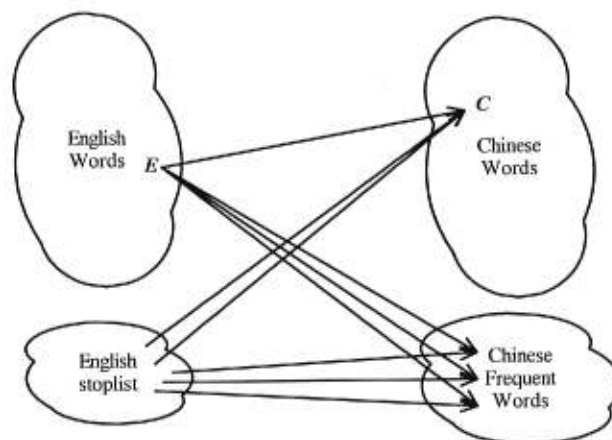


Figure 3.11: Effect of Chinese stop list in C-E translation.

It is more complicate to explain the second phenomenon. What we observe is that not removing the English stop-list words seems to help improving the precision when translating from English to Chinese. In fact, as illustrated in Fig. 3.11, the existence of the English stop list has two effects on the probability of the translation  $E \rightarrow C$ :

- 1 They may be often found to be together with the Chinese word  $C$ . As the result of the Expectation Maximization algorithm, the probability of  $E \rightarrow C$  may be reduced.
- 2 On the other hand, there are greater chances that they are found to be together with the most frequent Chinese words. Here we use the term “most frequent

words” instead of “Chinese stop-list” because even if the stop-list is applied, there may still remain some common words that has the same function as the stop-list words. The coexistence of English and Chinese frequent words reduces the probability that the Chinese frequent words are the translations of  $E$  and thus raised the probability of  $E \rightarrow C$ .

words	model4 (with no stop lists)			model3 (with English stop list)		
a.m.	f	的	0.146161	f	的	0.185531
access	f	的	0.158034	f	的	0.252953
adaptation	t	适应	0.171756	f	的	0.185532
add	t	补充	0.289871	f	的	0.191875
adopt	t	采用	0.21532	f	的	0.239604
agent	t	代理人	0.169783	f	的	0.223364
agree	t	同意	0.32433	t	同意	0.230816
airline	t	航空公司	0.30955	t	航空公司	0.263289
amendment	t	修订	0.284755	t	修订	0.247788
appliance	f	的	0.146441	f	的	0.191568
apply	t	适用	0.167856	f	的	0.234683
attendance	t	列席	0.158967	f	的	0.185532
auditor	f	的	0.146167	f	的	0.185813
average	t	平均	0.430765	t	平均	0.350076
base_on	t	根据	0.187724	f	的	0.270669
block	f	的	0.148584	f	的	0.214751
bottom	f	的	0.146328	f	的	0.190691
break_law	f	的	0.146161	f	的	0.185532
breath	t	呼气	0.309547	t	呼气	0.256398
briefing	t	简报	0.171755	f	的	0.185952
building	f	的	0.148937	f	的	0.227916
business	f	的	0.15169	f	的	0.220534
carry	t	进行	0.182572	f	的	0.262795
category	f	的	0.146164	f	的	0.185829
census	f	的	0.146165	f	的	0.185669

Figure 3.12: More E-C translations.

The second effect is believed to be more significant than the first one and it results in that model 2 has better precision than model 1 in English-Chinese translation. The above discussion also applies to Chinese-English translation.

To further prove the validity of our explanation, we compare the translations given by model 4 and model 3 (for English-Chinese translation) in Fig. 3.12. Based on the above theory, model 3 is expected to be the worst model for English-Chinese translation since it keeps the English stop-list and removes the Chinese stop-list which could balance out the effect of the English frequent words. The fact is that the results of model 3 ARE worse than model 4. Most words (a lot more than those of model 4) are translated to “的”, which might be the most frequent word of Chinese.

From the above, we may conclude that for better precision, only the stop-list of the target language should be applied in the model training.

### Analysis of Errors

The performance of the translation model is affected by many factors. Needless to say, the size and quality of the corpus and the precision of the alignment are very important. Besides, the following problems appear in the errors.

- 1 Some words in the source language do not have single-word translations in the target language. Examples are:

自由民主党 → liberal democratic party

新版 → new version

auditor → 审计员

Sometimes only part of the translations are given by the model. Being incomplete, this kind of translations is still useful for CLIR purpose. They are better than incorrect translations.

- 2 Correct translations of some words are in the stop-list of the target language. For example, “年” (year) is in the Chinese stop-list. As the result, we cannot have correct translation for “year”. Instead, “今年” (this year) or “每年” (every year) are returned.
- 3 Some translations are affected by the domain of the corpus. A large portion of the corpus was from the information site of Hong Kong government including

a lot of congress debates. Some translations reflecting the domain effects are:

mr → 议员 (congressman)

miss → 议员 (congressman)

house → 内务 (internal affair)

### 3.5 Summary

We went through the pre-training, training, and evaluation process of English-Chinese translation model in this chapter.

Prior to training, a series of procedures are carried to convert the raw parallel texts into sentence-level mappings. Besides Chinese coding conversion, Chinese word segmentation and English expression extraction, the most critical and challenging step is the alignment of sentences. A hybrid alignment method considering both sentence length and cognates (HTML markups) proved to be effective in this case.

The general idea of training the translation model is to learn the probability,  $p(t|s)$ , of having a word  $t$  in the translation of a sentence containing a word  $s$ . The resulting model showed encouraging precision in translating evaluation words. Stop-lists were found to have significant effect on model performance.

The models were only evaluated by translating single words. We will look into their performance in IR query translation in the next chapter.

# Chapter 4

## CLIR Experiments

In the last chapter, we described the process of training the translation model and inspected its precision by examining the evaluation lexicons. Encouraged by the results, we further apply the translation model in query translation of cross-language information retrieval. Experiments in both directions (E-C and C-E) are carried out on TREC collections using the SMART information retrieval system [Buc85]. The results are compared to those of mono-lingual IR, CLIR using MT systems, CLIR using dictionary, as well as English-French CLIR.

### 4.1 CLIR Approaches

Cross-language information retrieval receives query in one language and returns relevant documents in another language (or other languages). One scenario of using CLIR is when a surfer sends a query in Chinese to some Web search engine, he/she gets the related Web pages in Chinese, English, French, or Japanese ... . There's no need to translate the query by him(her)self to each language.

Various CLIR approaches were researched. Either the query or the documents [DLL96] are translated to other languages to achieve cross-language purpose. Here we look into query translation approaches which are more feasible than translating all the documents in collections.

Besides query translation using statistical model which is specifically studied in this work, machine translation systems and dictionary or terminology bases are also used. MT systems seem to be a straightforward tool for CLIR. However, Nie et al. [NSID99] argued that “MT and IR have widely divergent concerns”. An important concern of MT systems is to give syntactically correct translations which has little effect on IR performance since most IR implementations are based on single words in the query. Also, MT systems always select only one word in the target language among many possible choices that are synonyms or related words to the original word. This prevents a query expansion that may be naturally produced to improve IR performance. The last problem of applying MT systems in IR is that they are not available for many language pairs and difficult to build. Nie et al. [NSID99] also discussed the weakness of dictionary-based query translation which can not disambiguate between many possible translations in different contexts.

In the last chapter, we introduced the general idea of statistical translation model: the model learns (from a large set of alignments) the probability,  $p(t|s)$ , of having a word  $t$  in the translation of a sentence containing a word  $s$ . For an input sentence, the model then calculates a set of words that are most probable in its translation. One advantage of using statistical model is that it takes much less effort to establish a translation model than MT systems do. It is possible to explore the Web for parallel texts for many language pairs as the training material. From the IR point of view, statistical model has the advantage of being domain sensitive. A well-trained model is supposed to be able to appropriately choose related words based on the domain of the training material and the context of the source sentence. The experiments on English-French CLIR have shown encouraging results [NSID99]. In this chapter, we experiment the same approach for Chinese-English CLIR.

## 4.2 Chinese-English CLIR

In this section, we present the testing results for Chinese-English CLIR, in which queries are translated from Chinese into English to retrieve relevant documents in a

English collection.

### 4.2.1 The Collection

The English experiment collection we used is the AP collection in Trec6 and Trec7. 25 topics are provided in Trec6 while there are 28 new topics for Trec7. To conduct CLIR experiments, we manually translated the 25 English topics in Trec6 into Chinese. The Chinese queries are then translated by computer back to English.

Note that:

- 1 In Trec6 only 21 topics are provided relevant documents for evaluation. The following results are based on these 21 topics.
- 2 All the words in both the documents and queries are transferred to their citation forms.
- 3 The documents are indexed with the *ltn* weighting scheme of the Smart system. This is one of the forms of  $tf * idf$  weighting scheme and often used in IR. All the queries (either original or translated) that are not given weight before indexing are also indexed with the *ltn* scheme. All the translated queries with weights (given by the translation model) are indexed with the *mtc* scheme which is another variant of  $tf * idf$ . (See [Buc85] for formulas of the weighting schemes.)

### 4.2.2 Mono-Lingual IR Results

First of all, to provide a benchmark for CLIR results, we obtain a recall-precision average for mono-lingual IR. For the 21 English queries, an average precision of 0.3861 was obtained. We will compare the following results with this precision.

### 4.2.3 CLIR Using Translation Model

26 long queries of Trec6, including titles, descriptions, and narratives, are manually translated into Chinese. (Note that only queries in Trec6 are tested.) Before being



translated back to English, some common words that are useless for information retrieval are removed. Words such as 文件 (document), 有关 (relevant), and 无关 (irrelevant) frequently appear in the descriptions and narratives of the queries. They don't possess any information of queries and may decrease IR performance. Therefore, they are put into a stoplist.

The translation model is capable of translating sentences. The output is a series of words that are the most probable translations of the source sentence. Thus a necessary setting when using the translation model is the number of words we take from the output translation words. We introduce a parameter called *length factor* ( $C_{leng}$ ), which is the ratio of the lengths of target query and source query. For a query with  $N$  words, we take  $C_{leng}N$  words for its translation.

Another decision to make is whether we should use the weights given by the translation model for each word. These weights are actually the probabilities of having a word in the translation of the source query. Ideally, these weights should reflect the confidence level of the model on each word, as well as the context effect of the whole sentence. However, since the translation model is built upon a noisy corpus, we want to know if the weights really help with information retrieval. Therefore, for each following case, we test the cases using or not using the weights.

Tab. 4.1 lists the IR results of using the translation model to translate queries in Trec6. The performance is measured in terms of average precision - a standard measure in IR. It is also compared with that of the mono-lingual IR. Each query is passed to the model as a whole sentence. Different length factors are applied to investigate the effect of length factor. From the results we observe that:

- Generally, the IR precision for the machine translated queries are around 0.15, which is about 40% of mono-lingual IR.
- Using larger length factor (up to 3) results in better IR results. On the other hand, it is obviously not right to take too many words for each query. As we can see, when the length factor is 4, the precision starts to drop.

- For each length factor, the case using weight has better result. The larger the length factor is, the larger the difference between the two cases (using and not using weight) is. It is understandable since for a relatively long series of translations the words at the end have very low probabilities to be the translations of the source query. They should not be given the same weight as the preceding words.

Length Factor	Using Weight		Not Using Weight	
	Average Precision	%mono	Average Precision	%mono
1	0.1486	38.5%	0.1454	37.7%
1.5	0.1573	40.7%	0.1504	39.0%
2	0.1615	41.8%	0.1427	37.0%
3	0.1654	42.8%	0.1376	35.6%
4	0.1653	42.8%	0.1312	34.0%

Table 4.1: Results of CLIR using translation model (translate by queries).

One expectation to the statistical translation model is its ability of picking context sensitive translations among all possible translations. This is the main difference between a translation model and a dictionary-based approach. To investigate if the model really has this ability, we try to translate the query word by word, i.e., using the translation as a dictionary, and compare it with translating by whole query. Tab. 4.2 shows the IR results of queries translated in such way. For each word, 1, 2, or 3 translations are taken into the target query. We can see that when we expand the query enough (3 most probable translations per word), we can achieve similar IR precision as above. The IR performance of translating by sentence is not significantly superior than that of translating by word. There might be two reasons that caused this fact.

- Due to all the reasons such as the noisy nature of the training corpus and accuracy of text alignment, the probabilities learnt by the translation model

are not accurate enough to achieve the expected effect, i.e., choosing context-sensitive translation.

- Because the training corpus (collected from Web sites in Hong Kong) and the testing IR corpus (English collection of Trec6 and Trec7) are from different domains, a translation chosen by the model according to the domain feature of the training corpus may not be appropriate to the IR corpus.

Translations per word	Using Weight		Not Using Weight	
	Average Precision	%mono	Average Precision	%mono
1	0.1045	27.1%	0.1387	35.9%
2	0.1303	33.7%	0.1517	39.3%
3	0.1568	40.6%	0.1612	41.8%

Table 4.2: Results of CLIR using translation model (translate by words).

#### 4.2.4 CLIR Using Dictionary

It is interesting to compare the IR performance of the translation model with the dictionary-based approach. A medium-size Chinese-English dictionary, CEDICT [Den99], is adopted to translate the queries. The dictionary contains 23,509 entries. Each entry includes several English translations for a Chinese word. The queries are of course translated by word. Two possibilities are taking the first translation or all the translations in each entry. There's no weight given to the translations. The obtained IR results, as shown in Tab. 4.3, are similar to those of the translation model.

#### 4.2.5 Combining Translation Model and Dictionary

The experiment results so far showed that the IR performance of the translation model is not much better than that of a medium-size dictionary. While the translation model

	Average Precision	%mono
Take First Translation	0.1492	38.6%
Take All Translations	0.1530	39.6%

Table 4.3: Results of CLIR using a dictionary.

itself needs to be refined in various ways, we consider the possibility of combining the two approaches to expand query translations so as to improve the IR performance.

Here the query from the translation model is translated by query with the length factor of 1.5. The combination is actually implemented in the following two ways:

- 1 For each query, we take its translations by both the translation model and the dictionary. The probabilities given by the translation model are discarded. Instead we try to give different weights to the words in the two translations to find out a best combining ratio. (Tab. 4.4)

Combining Ratio (TM:DICT)	Average Precision	%mono
1:5	0.1890	49.0%
1:2	0.2371	61.4%
1:1	0.2583	66.9%
2:1	0.2424	62.8%
5:1	0.2023	52.4%

Table 4.4: CLIR Results of combining queries translated by translation model (without weight) and dictionary.

- 2 We first take the translation by the translation model with the given probabilities as the weights. The translated words by the dictionary are then added with a fixed weight. (Tab. 4.5)

Weight given to Dict. Translations	Average Precision	%mono
0.005	0.1933	50.1%
0.01	0.2233	57.8%
0.02	0.2284	59.2%
0.05	0.2058	53.3%

Table 4.5: CLIR Results of combining queries translated by translation model (keeping weight) and dictionary.

The results shown in Tab. 4.4 and Tab. 4.5 are encouraging. While applying alone, the two approaches showed similar IR performances as around 40%-mono. Combining them together in either of the above two ways, we achieved significant improvements. The best precision is high as 0.2583, which is 66.9%-mono.

The improvement shows that the translations given by the translation model and the dictionary complement each other well for IR purpose. The translation model may give either exact translations or incorrect but related words. Even though these words are not correct in the sense of translation, they are very possibly related to the subject of the query and thus helpful for IR purpose. The dictionary-based approach expands a query in another dimension. It gives all the possible translations for each word including those that are missed by the translation model.

### 4.3 English-Chinese CLIR

In the last section, we investigated the performance of the translation model in Chinese-English CLIR (in a English collection) and compared it with other query translation approaches. It is found that both the translation model and the Chinese-English dictionary CEDICT can achieve similar IR precision. However, combining the translated queries by these two approaches resulted in significant improvement. Now we are interested in if we can observe similar effect in the other direction, English-Chinese CLIR.

### 4.3.1 The Collection

We conduct the English-Chinese CLIR experiments in the Chinese collection of Trec5 and Trec6. There are 54 queries in the 170M Chinese collection encoded in GB. The queries are given in both Chinese and English. The documents come from two major Chinese newspapers, *People's Daily* and *Xinhua News Agency*. In the cases presented next, the documents are indexed by the *ltc* weighting scheme. All the translated queries with the probabilities given by the translation model are indexed by *mtc*.

### 4.3.2 Mono-Lingual IR Results

Once again, to provide a benchmark for CLIR results, we obtain a recall-precision average for Chinese mono-lingual IR. For the 54 queries, an average precision of 0.3976. All the following results will be compared to this precision.

### 4.3.3 CLIR Using Translation Model

Leng. Factor	Average Precision (%mono) Using Weight		Average Precision (%mono) Not Using Weight	
	seg	all	seg	all
1	0.1467 (36.9%)	0.1538 (38.7%)	0.1745 (43.9%)	0.1492 (37.5%)
1.5	0.1475 (37.1%)	0.1580 (39.7%)	0.1841 (46.3%)	0.1520 (38.2%)
2	0.1530 (38.5%)	0.1591 (40.0%)	0.1809 (45.5%)	0.1405 (35.3%)

Table 4.6: Results of CLIR using translation model.

The 54 English queries are translated to Chinese by the translation model. Before translation, all the words are converted to their citation forms. The common words are eliminated through a stop list.

In the last section, queries are passed to the translation model in two ways, all in one sentence or word by word. This time we compare translating by query to

translating by segment. By segments we mean the different fields of a query, such as *title*, *description*, and *narrative*.

Tab. 4.6 lists all the CLIR results using the translation model. “Seg” and “all” refer to translate by segment or query, respectively. Different length factors (as explained in the last section) are applied. For each length factor, we try using or not using the probabilities given by the model, as in the last section.

First of all, we notice that the translation model has about the same performance as in Chinese-English CLIR. Generally, around 40% of mono-lingual IR precision can be achieved. In the following analysis, we further look into the effects of different ways of applying the translation model.

- Length factor – Similarly to what we found in the last section, appropriately expanding the translated query results in better precision. To some extent, IR precision increases with length factor.
- Using weight or not – We observe that for all the cases we translate by query, using weight gives better results than not using weight. It’s the same as in the Chinese-English cases where we also translate by query. However, if the queries are translated by segment, not using the weights results in much better results. As we argued in the last section, words at the end of a relatively long query have very low probabilities and should not be treated the same as the preceding words. In the cases of translating by segments, translated queries are relatively short and the probabilities of all the words are relatively close. The inaccurate nature of these weights has dominant effect. It is then better not using them.
- Translating by segment or query – While using weight, we can see translating by query is better than translating by segment. It is the opposite case when weights are not used. For each length factor, the best precision occurs when we translate by segment without using weight.

#### 4.3.4 CLIR Using Dictionary

An online English-Chinese dictionary [Ano99b] is adopted to test the dictionary-based approach. Tab. 4.7 shows the CLIR results. We obtain precision that is slightly lower than those of the translation model.

	Average Precision	%mono
Take First Translation	0.1171	29.5%
Take All Translations	0.1427	35.9%

Table 4.7: Results of CLIR using an online English-Chinese dictionary.

#### 4.3.5 Combining Translation Model and Dictionary

For the same reason as in Chinese-English CLIR, we combine queries translated by translation model and dictionary. Two strategies of combination as described in the last section are adopted. The query from the translation model is translated by segment with the length factor as 1.5.

Combining Ratio (TM:DICTIONARY)	Average Precision	%mono
1:2	0.1898	47.7%
1:1	0.2125	53.4%
2:1	0.2232	56.1%
3:1	0.2208	55.5%

Table 4.8: CLIR Results of combining queries translated by translation model (without weight) and dictionary.

From Tab. 4.8 and Tab. 4.9 we find that combining translation model and dictionary results in significant improvement of CLIR precision, as it does in English-Chinese CLIR. Interestingly, in Tab. 4.8 the best precision occurs when the combining



Weight given to Dict. Translations	Average Precision	%mono
0.001	0.1638	41.2%
0.005	0.1874	47.1%
0.01	0.1784	44.9%
0.02	0.1648	41.1%

Table 4.9: CLIR Results of combining queries translated by translation model (keeping weight) and dictionary.

ratio is 2:1, while in C-E CLIR (Tab. 4.4) the best precision was obtained with the ratio as 1:1. From Tab. 4.10, it seems reasonable to say that the best combining ratio depends on the CLIR performance of using translation model and dictionary alone. If they have similar performance, a “1:1” ratio is appropriate, otherwise we should “trust” more the one with better performance.

	Translation Model	Dictionary	Best Comb. Ratio	Best Precision
E-C CLIR	0.1841	0.1427	2:1	0.2232
C-E CLIR	0.1504	0.1530	1:1	0.2583

Table 4.10: Best combining ratios for E-C and C-E CLIR.

### 4.3.6 CLIR Using MT System

One advantage of parallel text based translation model is that it is easier to establish than MT systems. Now that we have examined the CLIR performance of the translation model, we will compare it with two existing MT systems.

#### Sunshine WebTran Server

Using the Sunshine WebTran server [Ano99c], an online English-Chinese MT system, to translate the 54 English queries, we obtained an average precision as 0.2001, which

is 50.3% of the mono-lingual precision. The precision is higher than that of using the translation model (0.1804) or the dictionary (0.1427) alone but lower than the precision of using them together (0.2232).

### **Transperfect**

Kwok [Kwo99] investigated the CLIR performance of an English-Chinese MT software called Transperfect, using the same TREC Chinese collection as we used in this work. By using the MT software alone, Kwok achieved 56% of mono-lingual precision. The precision is improved to 62%mono by refining translation with dictionary. Kwok also adopted pre-translation query expansion, which further improved the precision to 70%mono.

The best E-C CLIR precision using the translation model (and dictionary) is 56.1%mono. It is lower than what Kwok achieved using MT system. However, the difference is not significant. In addition, we did not apply the same refinement measure as Kwok did to achieve 70% performance of the mono-lingual IR. It is possible that we would obtain some improvement with the similar refinement. Unfortunately, we do not have the same E-C translation system to compare it with in more details.

## **4.4 Comparison to English-French CLIR**

Preceding to this work, Nie et al. worked on English-French CLIR using translation model trained by Web parallel corpus [NSID99]. We are interested in the difference of applying the same framework on different language pairs.

The experiments of [NSID99] were conducted on the English AP and French SDA corpora in TREC6 and TREC7. The performance of the translation model was compared with that of mono-lingual IR, MT system, and dictionary-based translation. A series of recent results are shown in Fig. 4.1. The MT system was found to be able to perform query translation with reasonable quality. The translation model gave less but still comparable precision to the MT System-Systran. It out-performed the dictionary-based approach. It was also found that combining dictionary appropriately

with the translation model can considerably improve the average precision. In comparison with a model trained with a manually established parallel corpus (Hansard), we see that the model based on Web documents achieves comparable performance. This result is encouraging. It shows that the Web parallel texts may substitute a manual parallel corpus for CLIR.

		Trec6	Trec7
Mono-lingual IR	E-E	0.2865	0.3202
	F-F	0.3686	0.2764
CLIR using MT	F-E (%mono)	0.3098 (107.0%)	0.3293 (102.8%)
	E-F (%mono)	0.2727 (74.0%)	0.2327 (84.2%)
CLIR using dictionary	F-E (%mono)	0.1707 (59.0%)	0.1701 (53.1%)
	E-F (%mono)	0.2305 (62.5%)	0.1352 (48.9%)
CLIR using trans. model	F-E (%mono)	0.2389 (82.5%)	0.3146 (98.3%)
	E-F (%mono)	0.2504 (67.9%)	0.2289 (82.8%)
CLIR using Hansard model	F-E (%mono)	0.2166 (74.8%)	0.3124 (97.6%)
	E-F (%mono)	0.2501 (67.9%)	0.2587 (93.6%)

Figure 4.1: English-French CLIR results.

It is obvious that the Chinese-English translation model has far lower CLIR performance than that of the English-French model established with the same method. A main reason lies in the greater difference between English and Chinese than that between English and French. This problem exists in many phases of this work, from text alignment to query translation.

Some of the problems affecting CLIR precision are as follows.

- The Web-collected corpus is noisy and it is difficult to align English-Chinese texts. The currently used alignment method has poorer performance than that in English-French alignment. This then leads to poorer performance of the translation model.
- In general, we observe higher variability in Chinese-English translation than in English-French translation, which affects precision of translation model.

- For E-C CLIR, although queries in both languages are provided, the English queries were not strictly translated from the original Chinese ones. For example, 人权状况 (*human right situation*) was translated into *human right issue*. We can not expect the translation model to translate *issue* back to 状况 (*situation*).
- The training source and the collections are from different domains. The Web corpus are retrieved from the parallel sites in Hong Kong while the Chinese collection is from *People's Daily* and *Xinhua News Agency* which are newspapers in mainland China. As the result, some important political terms, abbreviations, and proper nouns in the collection are not known by the model. Examples are 最惠国 (*most-favored-nation*) and 一国两制 (*one-nation-two-systems*).

All the above factors may explain the difference between C-E and E-F CLIR.

## 4.5 Summary

In this chapter, we first briefly discussed different CLIR query translation approaches. Then we presented experiment results of E-C and C-E CLIR. In each direction, we tried applying the translation model in various ways, i.e., using different length factors; using or not using the weight provided by the model; translating by word, segment or query. Analysis on results of these options were given.

It is found that using the translation model alone we can obtain around 40% of mono-lingual IR precision. We also tested dictionary-based translation on both directions with the results no better than those of the translation model. Interestingly, the translations given by the translation model and dictionary approach seem to complement each other well. By combining them together, IR precision were significantly improved to 66.9% (C-E) and 56.1%(E-C).

We also compared the translation model with two English-Chinese MT systems. The Sunshine WebTran system turned out to be better than the translation model alone and poorer than combining the translation model with dictionary. With the help of dictionary and using query expansion, Kwok achieved 70%mono with the

Transperfect MT system. The results we obtained with the translation model are comparable with those systems.

Due to the greater difference between the two languages, the English-Chinese translation model doesn't have the same CLIR performance as the English-French model. We pointed out some existing problems.

# Chapter 5

## Conclusions

Three topics, parallel text mining, statistical translation model training, and cross-language information retrieval, are involved in this thesis. Originating from the need for a low-cost and effective query translation tool for CLIR, we chose statistical translation model which has to be trained by large-scale parallel corpora. However, there is only a small number of parallel corpora available. The text mining system described in this thesis aims to find a large amount of parallel texts from the web to feed the training process of the translation model.

The difficulty of Web parallel text mining lies in how we could find parallel Web pages among zillions of others without having to read all of them. We proposed a heuristic method which finds candidate bilingual sites by taking advantage of Web search engines, retrieves URLs from candidate sites by host crawling, and then scans parallel pairs according to naming patterns. To search efficiently, the implementation of the algorithm (PTMiner) adopted a distributed model so that there can be arbitrary number of “miners” working simultaneously on multiple bilingual sites. With the PTMiner system, we successfully built a large-scale Chinese-English parallel corpus with acceptable accuracy. Currently we filter the mined pairs by using language identification tool and comparing text lengths. More effective methods are needed to further improve the accuracy of the corpus.

For the translation model, we adopted the statistical model proposed by Brown

et al. [BPPM93] and its implementation by the RALI group. Our work is focused on the preprocessing of the raw parallel texts so that they can be fed into the training process of the translation model. In this pre-training process, we conducted several operations on the parallel corpus: Chinese coding conversion, Chinese-English parallel text alignment, Chinese word segmentation and English expression extraction.

One of the pre-training operations is sentence alignment. The method we used [SFI92] was originally designed for European languages in which cognates (similar words) are common. This is not the case for Chinese-English alignment. However, because the HTML markups exist in both English and Chinese texts, we use them as cognates. Our experiments showed that this is an effective way to align Chinese-English texts in HTML.

With a set of randomly selected words, we tested the precision of the translation model. It is shown that about 80% of the most probable translations are correct. This precision is reasonable. Finally, we applied the translation model in CLIR query translation. The translation model showed similar performance and phenomenon on E-C and C-E CLIR. As the model alone could achieve limited IR precision, we improved the precision significantly by combining the translations by both the translation model and dictionaries. The results are comparable with those of some MT systems.

Compared to the similar frame work of English-French CLIR using the translation model trained by Web corpus, the Chinese-English CLIR results are poorer because of the much larger difference between Chinese and English than that between English and French. Despite this inherited factor, we believe this work can be improved at many points along the chain of operations. Some possible improvements are:

- In parallel text mining, we believe more sophisticated evaluation scheme could be used to improve the precision of the result corpus. Besides length difference and language identification, we expect a reliable way to filter by HTML structure similarity and alignment confidence value.
- In translation model training, improvement could be achieved by refining the

sentence alignment method. Two possible ways are: modify the definition of cognates to better recognize HTML markups, and incorporating a small lexicon to help with alignment.

- In CLIR, we could consider applying some query expansion techniques to include synonyms. The indexing stoplist could also be refined to exclude more common functional words that do not help with IR.



# Bibliography

- [Ano99a] Anonymous. New search engine to snare all the Web. <http://techweb.com/wire/story/TWB19990809S0002>, August 1999.
- [Ano99b] Anonymous. Sunrain.net - English-Chinese dictionary. [http://sunrain.net/r\\_eedict\\_e.htm](http://sunrain.net/r_eedict_e.htm), 1999.
- [Ano99c] Anonymous. Sunshine WebTran server. <http://www.readworld.com/translate.htm>, 1999.
- [BDH<sup>+</sup>94] C. M. Brown, B. B. Danzig, D. Hardy, U. Manber, and M. F. Schwartz. The harvest information discovery and access system. In *Proc. 2nd International World Wide Web Conference*, 1994.
- [BLM91] P. F. Brown, J. C. Lai, and R. L. Mercer. Aligning sentences in parallel corpora. In *29th Annual Meeting of the Association for Computational Linguistics*, pages 89–94, Berkeley, Calif., 1991.
- [BPPM93] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311, 1993.
- [Bro98] R. D. Brown. Automatically-extracted thesauri for cross-language IR: When better is worse. In *1st Workshop on Computational Terminology (Computerm)*, pages 15–21, 1998.

- [BSY95] M. Balabanovic, Yoav Shoham, and Y. Yun. An adaptive agent for automated web browsing. *Journal of Visual Communication and Image Representation*, 6(4), 1995.
- [Buc85] C. Buckley. Implementation of the SMART information retrieval system. Technical Report 85-686, Cornell University, 1985.
- [Ca91] J. S. Chang and al. Chinese word segmentation through constraint satisfaction and statistical optimization. In *ROCLING 4*, pages 147–165, 1991.
- [Che93] S. F. Chen. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31th Annual Meeting of the Association for Computational Linguistics*, pages 9–16, Columbus, Ohio, 1993.
- [Chu93] K. W. Church. Char.align: A program for aligning parallel texts at the character level. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 1–8, Columbus, Ohio, 1993.
- [CK92] K. J. Chen and S. H. Kiu. Word identification for Mandarin Chinese sentences. In *5th International Conference on Computational Linguistics*, pages 101–107, 1992.
- [CMS97] R. Cooley, B. Mobasher, and J. Srivastava. Web Mining: Information and pattern discovery on the World Wide Web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, November 1997.
- [CN00a] Jiang Chen and Jian-Yun Nie. Automatic construction of parallel English-Chinese corpus for cross-language information retrieval. In *6th Applied Natural Language Processing Conference*, pages 21–28, Seattle, Washington, May 2000.
- [CN00b] Jiang Chen and Jian-Yun Nie. Parallel Web text mining for cross-language IR. In *Content-Based Multimedia Access - RIAO'2000*, pages 62–77, Paris,

France, April 2000. Centre de Hautes Etudes Internationales d'informatique documentaire (CID), Collège de France.

- [CN00c] Jiang Chen and Jian-Yun Nie. Web parallel text mining for Chinese-English cross-language information retrieval. In *2000 International Conference on Chinese Language Computing (ICCLC2000), Workshop on Virtual University for Multilingual Education*, Chicago, IL, USA, July 2000.
- [Den99] Paul Denisowski. Cedict (chinese-english dictionary) project. <http://www.mindspring.com/~paul.denisowski/cedict.html>, 1999.
- [DEW96] R. B. Doorenbos, O. Etzioni, and D. S. Weld. A scalable comparison shopping agent for the World Wide Web. Technical Report Technical Report 96-01-03, Dept. of Computer Science and Engineering, University of Washington, 1996.
- [DLL96] S.T. Dumais, T.K. Landauer, and M.L. Littman. Automatic cross-linguistic information retrieval using latent semantic indexing. In *SIGIR'96 Workshop on Cross-Linguistic Information Retrieval*, 1996.
- [FM94] Pascale Fung and Kathleen McKeown. Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping. In *AMTA 94: Partnerships in Translation Technology*, pages 81–88, Columbia, Maryland, October 1994.
- [For95] Coping with complex data, the Forrester Report. Technical report, Forrester Research, Inc., April 1995.
- [Fun] Pascale Fung. A statistical view on bilingual lexicon extraction: from parallel corpora to non-parallel corpora. In *Lecture Notes in Artificial Intelligence*, volume 1529, pages 1–17. Springer Publisher.
- [GC91] William A. Gale and Kenneth W. Church. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Meeting of*

- the Association for Computational Linguistics*, pages 177–184, Berkeley, Calif., 1991.
- [Han97] J. Han. OLAP mining: An integration of OLAP with data mining. In *Proc. 1997 IFIP Conference on Data Semantics (DS-7)*, pages 1–11, Leysin, Switzerland, Oct. 1997.
- [Han99] J. Han. Data mining. In J. Urban and P. Dasgupta, editors, *Encyclopedia of Distributed Computing*. Kluwer Academic Publishers, 1999.
- [HCC<sup>+</sup>97] J. Han, J. Chiang, S. Chee, J. Chen, Q. Chen, S. Cheng, W. Gong, M. Kamber, K. Koperski, G. Liu, Y. Lu, N. Stefanovic, L. Winstone, B. Xia, O. R. Zaiane, S. Zhang, and H. Zhu. DBMiner: A system for data mining in relational databases and data warehouses. In *Proc. CASCON'97: Meeting of Minds*, Toronto, Canada, November 1997.
- [IFP97] P. Isabelle, G. Foster, and P. Plamondon. SILC: un système d'identification de la langue et du codage. <http://www-rali.iro.umontreal.ca/ProjetSILC.en.html>, 1997.
- [IMT99] Intelligent Miner for Text. <http://www.software.ibm.com/data/iminer/fortext/about.html>, 1999.
- [Jin92] Wanying Jin. A case study: Chinese segmentation and its disambiguation. Technical Report MCCS-92-227, Computing Research Laboratory, New Mexico State University, Las Cruces, 1992.
- [Jin94] Wanying Jin. Chinese segmentation disambiguation. In *Proceedings of the International Computational Linguistics-94 (COLING'94)*, pages 1245–1249, Japan, 1994.
- [KR93] M. Kay and M. Röscheisen. Text-translation alignment. *Computational Linguistics*, 19:121–142, 1993.

- [KS95] D. Konopnicki and O. Shmueli. W3QS: A query system for the World Wide Web. In *Proc. of the 21th VLDB Conference*, pages 54–65, Zurich, 1995.
- [Kwo99] K. L. Kwok. English-chinese cross-language retrieval based on a translation package. In *Workshop of Machine Translation for Cross Language Information Retrieval, Machine Translation Summit VII*, Singapore, 1999.
- [Liy87] Y. Q. Liy. Difficulties in CHINESE language processing and method to their solution. In *Proc. of 1987 International Conference on Chinese Information Processing*, volume 2, pages 125–126, 1987.
- [Lun99] Ken Lunde. *CJKV Information Processing*. O’Reilly, January 1999.
- [LZ91] N. Y. Liang and Y. B. Zhen. A Chinese word segmentation model and a Chinese word segmentation system PC-CWSS. In *COLIPS’91*, volume 1, pages 51–55, 1991.
- [MCD98] Bill McCarty and Luke Cassady-Dorion. *Java Distributed Objects*. Sams, 1998.
- [MJHS96] B. Mobasher, N. Jain, E. Han, and J. Srivastava. Web mining: Pattern discovery from World Wide Web transactions. Technical Report Technical Report TR 96-050, Dept. of Computer Science, University of Minnesota, Minneapolis, 1996.
- [NJH94] Jianyun Nie, Wanying Jin, and M. L. Hannan. A hybrid approach to unknown word detection and segmentation of chinese. In *International Conference on Chinese Computing*, pages 326–335, Singapore, 1994.
- [NRB95] Jianyun Nie, Xiaobo Ren, and Martin Brisebois. A unifying approach to segmentation of Chinese and its application to text retrieval. In *ROCLING 8*, pages 172–190, August 1995.

- [NSID99] Jianyun Nie, Michel Simard, Pierre Isabelle, and Richard Durand. Cross-language information retrieval based on parallel texts and automatic mining parallel texts from the Web. In *ACM SIGIR'99*, pages 74–81, August 1999.
- [PE95] M. Perkowitz and O. Etzioni. Category translation: learning to understand information on the internet. In *Proc. 15th International Joint Conference on AI*, pages 930–936, Montreal, Canada, 1995.
- [Res98] Philip Resnik. Parallel stands: A preliminary investigation into mining the Web for bilingual text. In *AMTA'98*, October 1998.
- [Sem98] Text mining and the knowledge management space. Technical report, Semio Corporation, 1998.
- [SFI92] Michel Simard, George F. Foster, and Pierre Isabelle. Using cognates to align sentences in bilingual corpora. In *Proceedings of TMI-92*, Montreal, Quebec, 1992.
- [SM83] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
- [SP98] Michel Simard and Pierre Plamondon. Bilingual sentence alignment: Balancing robustness and accuracy. *Machine Translation*, 13:59–80, 1998.
- [SS91] R. Sproat and C. Shih. A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, 4(4):336–351, 1991.
- [Tka98] Daniel Tkach. Text mining technology – turning information into knowledge – a white paper from IBM. Technical report, IBM Software Solutions, February 1998.
- [Uni99] The Unicode standard – a technical introduction. <http://www.unicode.org/unicode/standard/principles.html>, 1999.

- [Wu94] Dekai Wu. Aligning a parallel English-Chinese corpus statistically with lexical criteria. In *ACL-94: 32nd Annual Meeting of the Assoc. for Computational Linguistics*, pages 80–87, Las Cruces, NM, June 1994.
- [Wu95] Dekai Wu. Large-scale automatic extraction of an English-Chinese lexicon. *Machine Translation*, 9(3-4):285–313, 1995.
- [WX94] Dekai Wu and Xuanyin Xia. Learning an English-Chinese lexicon from a parallel corpus. In *Proceedings of AMTA94*, pages 206–213, Columbia, Maryland, October 1994. Association for Machine Translation in the Americas.
- [ZH98] O. Zaiane and J. Han. WebML: Querying the World-Wide Web for resources and knowledge. In *Proc. (CIKM'98) Int'l Workshop on Web Information and Data Management (WIDM'98)*, pages 9–12, Bethesda, Maryland, November 1998.
- [ZXH98] O. R. Zaiane, M. Xin, and J. Han. Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs. In *Proc. Advances in Digital Libraries Conf. (ADL'98)*, pages 19–29, Santa Barbara, CA, April 1998.