MINING PARALLEL DOCUMENTS USING LOW BANDWIDTH AND HIGH PRECISION CLIR FROM THE HETEROGENEOUS WEB

by

SHI, YUE

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirement for
the Degree of Master of Philosophy
in the Department of Electronic and Computer Engineering

August 2011, Hong Kong
Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

SHI, YUE

August 2011
MINING PARALLEL DOCUMENTS USING LOW BANDWIDTH AND HIGH PRECISION CLIR FROM THE HETEROGENEOUS WEB

by

SHI, YUE

This is to certify that I have examined the above MPhil thesis and have found that it is complete and satisfactory in all respects, and that any and all revisions required by the thesis examination committee have been made.

Prof. Pascale Fung, Thesis Supervisor

Prof. Weichuan Yu, Chairman

Prof. Raymond Chi-Wing Wong, Thesis Committee Member

Prof. Ross Murch, Department Head

The Department of Electronic and Computer Engineering

August 2011
ACKNOWLEDGMENTS

I would like to express my deep gratitude to my supervisor, Prof. Pascale Fung, who has given me a lot of advice and kindly support in my research during two years of my MPhil study.

I would like to thank the Department of Electronic and Computer Engineering and Human Language Technology Center (HLTC) for providing me study and research facilities.

Thanks to Emmanuel Prochasson, research associate of HLTC to help me on completing the research work.

Also, I would like to thank other the postgraduate students, research assistants and research associates in HLTC who have given me a lot of help and valuable opinions in different ways.

Last, but not the least, thanks to my parents for their constant love and support they have always given me.
# TABLE OF CONTENTS

CHAPTER 1 Introduction ..................................................................................1
  1.1 Mining Parallel Resource from the Web .................................................. 1
  1.2 Objectives ............................................................................................... 2
  1.3 Organization of the Thesis ...................................................................... 4

CHAPTER 2 Background ..................................................................................5
  2.1 Parallel Resources on the Web ............................................................... 5
  2.2 Approaches to Mine Parallel Documents ..............................................10
      2.2.1 Structure Based Approaches .......................................................... 10
      2.2.2 Content Based Approaches ............................................................ 11

CHAPTER 3 Our Content Based Approach Supplemented with Structural
  Information ..................................................................................................15
  3.1 Source Document Representation ..........................................................16
  3.2 Search Target Documents with Search Query Relevance Score (SQRS)........18
  3.3 Document Verification ..........................................................................25
      3.3.1 Dynamic Time Warping (DTW) Score ............................................25
      3.3.2 R² Regression ................................................................................27
      3.3.3 Combining DTW and R² .................................................................28
      3.3.4 Structural Features .......................................................................29
  3.4 Improve Recall by Structure Based Information ....................................29

CHAPTER 4 Experiment Setup and Result ......................................................31
4.1 Experiment setup........................................................................................................31
  4.1.1 Website Parsing..................................................................................................31
4.2 Content Based Baseline..........................................................................................32
4.3 Experiment 1: Find Target Documents..................................................................33
4.4 Experiment 2: Parallel Sentence Extraction for SMT .........................................36
4.5 Experiment 3: Bilingual Lexicon Extraction.......................................................38
4.6 Experiment 4: Parallel Document Mining for Hindi-English...............................38
4.7 System Performance and Scalability......................................................................39
4.8 Future Works and Discussion................................................................................40

CHAPTER 5 Conclusion.................................................................................................42

References .....................................................................................................................44

APPENDIX A System Documentation ...........................................................................49

1. System Requirement................................................................................................49
2. Package Content......................................................................................................49
3. Installation ................................................................................................................52
4. Commands of the Tool............................................................................................53
5. Database Structure..................................................................................................54
6. Database Operation................................................................................................56
LIST OF TABLES

Table 3.1 SQRS of Query 1 ................................................................. 21

Table 3.2 SQRS of Query 2 ................................................................. 21

Table 3.3 Result quality and SQRS ..................................................... 22

Table 3.4 Comparison of target document found with and without SQRS .......... 25

Table 3.5 DTW and precision of candidate pairs ........................................ 27

Table 3.6 Mining precision of DTW and R^2 ........................................... 29

Table 3.7 Comparison of content based method and URL matching ............... 30

Table 3.8 Source documents for pure content based approach ....................... 30

Table 4.1 Comparison of different methods ............................................. 35

Table 4.2 Output document pairs of Experiment 1 ..................................... 36

Table 4.3 BLEU score obtained for SMT (Experiment 2) .............................. 38

Table 4.4 Result of Experiment 4 .......................................................... 39
LIST OF FIGURES

Figure 2.1 Example of parallel website.................................................................6

Figure 2.2 Example of comparable websites (Reuters) ...........................................8

Figure 2.3 Example of quasi-comparable websites...............................................9

Figure 2.4 Typical content based approach used in previous works.......................12

Figure 2.5 Basic steps of content based approach using search engine APIs..........13

Figure 3.1 Our content based approach .................................................................16

Figure 3.2 Search result of Query 1 (a) and 2 (b) on Google.com.........................20

Figure 3.3 Flowchart of query expansion algorithm............................................24

Figure 3.4 DTW of parallel and non-parallel document pairs................................26

Figure 3.5 R² of parallel and non-parallel document pairs.....................................28
MINING PARALLEL DOCUMENTS USING LOW BANDWIDTH AND HIGH PRECISION CLIR FROM THE HETEROGENEOUS WEB

SHI, YUE

Department of Electronic and Computer Engineering

ABSTRACT

In this thesis, we propose a content-based method of mining bilingual parallel documents from websites that are not necessarily related to each other.

Parallel corpora are a key resource as training data for statistical machine translation, and for building or extending bilingual lexicons and terminologies. There are two existing approaches for automatically mining parallel documents from the web, structure based approach and content based approach. Structure based methods work only for parallel websites and most of the content based methods either require large scale computational facilities, network bandwidth or not applicable to heterogeneous web.

We suggest that parallel documents can be mined with high precision from websites that are not necessarily parallel to each other. We propose a novel content based method using cross lingual information retrieval (CLIR) with query feedback and verification and supplemented with structural information, to mine parallel resources
from the entire web using search engine APIs. The method goes beyond URL matching and hyper-links tracking to find parallel documents from non-parallel websites.

We introduce a Search Query Relevance Score (SQRS) to measure the translation quality and select keywords to generate queries for further mining of target documents. Our approach neither requires crawling all web documents in the target language, nor does it require machine translation of the full text. It therefore requires less bandwidth and computing resources. We obtained a very high mining precision (88%) on the parallel documents by the pure content based approach and improve the quantity by mining parallel websites using the structure based methods. After extracting parallel sentences from the mined documents and using them to train an SMT system, we found that the SMT performance, with a higher BLEU score, is comparable to that obtained with high quality manually translated parallel sentences illustrating the excellent quality of the mined parallel materials.
CHAPTER 1
INTRODUCTION

1.1 Mining Parallel Resource from the Web

Parallel corpora, which contain translated sentences, are essential resources as training data for statistical machine translation and building or extending bilingual lexicons and terminologies. Parallel resources such as bilingual lexicon and sentence translations are typically obtained from translated parallel documents.

In the past, parallel resources from human translations and translations of newswire documents were the main source of parallel corpora for statistical machine translation systems. These kinds of resources are very costly and size limited. However, large amount of documents are publicly available on the World Wide Web under trillions of URLs\(^1\) in hundreds of languages. Thus, the web becomes an important source of parallel documents.

In previous studies, many researches have been done on parallel document mining from the web, using both structure based and content based methods. Previous approaches commonly require downloading all documents in both languages and then performing alignment locally. However, the number of documents on the web is growing so fast that it is not yet feasible for most research institutions to build a local archive.

\(^1\) *We knew the web was big*... on the Official Google Blog.
http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html.
containing trillions of documents. There is a need to readdress the problem of how to mine parallel documents from the entire web.

With the development of the search technology, online search engine becomes an efficient tool in finding documents on the web which is heterogeneous in nature. Online search engines such as Google or Bing have built a comprehensive and up-to-date index including almost all public documents on the Internet.

The use of the search engine is simple and fast which only includes one HTTP client server communication and search time for each query which is less than a second. It would be efficient and helpful for research institutions to mine parallel resources from the entire web if we can make use of the search engine APIs to find target document for any source document.

1.2 Objectives

Previous work by [1] [2] [3] and [4] produced parallel sentences mainly from pre-downloaded bilingual corpus in both source and target languages. It is very costly in CPU power if the size of the comparable corpus is large, for example, the entire web.

In the thesis, my research focuses on investigating a new method and implementing a tool for mining parallel resources from the web for training machine translation systems and bilingual lexicon extraction with high precision and low bandwidth.

(a) High Precision

We focus on mining directly translated (parallel) documents rather than comparable ones because the parallel resources are mainly used for statistical machine translation
(SMT) systems as training data and bilingual lexicon extraction. The quality of parallel documents must be high enough for these kinds of applications.

(b) Low Bandwidth

The web contains trillions of URLs. Building an archive of these URLs will consume a large amount of bandwidth, storage and CPU power. We want to establish a new method which can be used by most research institutions. So the method should not in the way of creating local archive of all target documents and then conducting alignment. Our novel approach uses online search engines to find, by keywords, the translation of each document.

Using the new method should also save bandwidth if we only find parallel document pairs for a small amount of source documents.

In previous works by [5] and [2], they proposed to align parallel documents in a pre-downloaded, closed collection of web documents. The bandwidth and CPU power utilization is very high for downloading documents from the entire web and aligning documents in the corpus.

(c) Improved Recall

The web, however, is heterogeneous in nature. We do not know whether the desired target document exists or not and where the document is. Our method must search the target document in the entire web scope without any restriction on domain name, language, date or etc.
There are a lot of documents containing the same keywords, on the same topics or are the translations of the given document. It is very challenging to search the web for a specific document in another language.

To mine documents from the entire web, the system should be scalable. There must not be any centralized process that limits the system to run only on single CPU core or single machine. The structure based methods, such as URL matching, hyper-links and HTML structure, introduced in previous works are simple and accurate for mining parallel documents from bilingual parallel websites compared to content based methods. We aim to use URL structure based methods as part of our approach to generate more parallel resources from parallel sites and use them as sources of expansion.

1.3 Organization of the Thesis

The organization of the thesis is as follows. We first introduce the parallel resources that exist on the World Wide Web and the approaches used in previous studies to mine parallel documents and sentences in CHAPTER 1 and CHAPTER 2 respectively.

Our novel content based approach is presented in CHAPTER 3. The experiments that prove the algorithm and the results, the parallel sentences extraction part and system performance and scalability are shown in CHAPTER 4.

Finally in the last chapter conclusions of the thesis and suggestions for future works are presented.
CHAPTER 2
BACKGROUND

2.1 Parallel Resources on the Web

Parallel resources are translated documents and sentences. They are mainly used as training data for a statistics machine translation system or bilingual lexicon extraction. On the World Wide Web, parallel resources reside on a diverse range of websites which can be classified into the following categories:

(a) Parallel Websites

A single website with structurally aligned translated pages is a parallel website. Typically there are websites of institutions, governments and commercial companies. These websites provide information in more than one language. The same pages in different languages usually have similar URLs. For example, as shown in Figure 2.1, Financial Times Chinese have the same news in English and Chinese and the two pages have similar URLs.
Airbus and Boeing call end to ‘duopoly’

By Mark Otell at the Paris Air Show

Airbus and Boeing have declared their almost total dominance of the market for large commercial aircraft is over.

Speaking on the first day of the Paris Air Show, Jim Albaugh, head of Boeing’s civil jet division, made the frank admission as Brazilian, Chinese, Canadian and Russian companies are all set to enter the 100-150 seat plus market with jets of their own in the next five years. “The days of the duopoly with Airbus are over,” he said.

Tom Enders, Airbus chief executive, agreed but added that the newcomers were not targeting the entire product range of either of the two dominant groups.

He told the Financial Times: “The duopoly is over in the 100 to 150 aircraft segment because this is where the new entrants… want to be – so that doesn’t mean the duopoly is over in the entire range of products.”

Figure 2.1 Example of parallel website

---

(b) Comparable Websites

Websites contain parallel contents in different languages without any structural relation in URL, hyperlinks or HTML structure between document pairs. Press agencies have independent content management systems and editors for publishing news in different regions and languages. That is, on comparable websites, parallel documents exist but they are not linked together by website editors.

For example, Reuters China (cn.reuters.com) and Reuters.com have independent editors and use identical content management systems. Some news is directly translated from the other site but there is no hyper-link to the original version. As shown in Figure 2.2, the two translated pages neither have similar URL nor are they hyper-linked to each other.
Figure 2.2 Example of comparables websites (Reuters)\(^3\)\(^4\)

(c) Quasi-comparable Websites

Independent websites that contain some translated parallel contents are quasi-comparable. They may contain stories, documentations and books chapters in many languages on different websites.

For example, in Figure 2.3, the book *Uncle Tom’s Cabin* (湯姆叔叔的小屋) is available in Chinese (a) and English (b). The Chinese version is hosted on *tianya* book online and the English version is on the University of Virginia Library. The two

\(^3\) English document at http://www.reuters.com/article/idUSN2332431020090423

第章 给读者介绍一位好心人

二月的某一天，天气依然比较寒冷。黄昏时分，正在一场庆祝活动开展的餐厅里，两位绅士相对而坐，喝着酒。他们没有服务生在旁边侍候，他们夹着酒瓶，好像在商量什么重要的事情。

为了便适用于读者阅读，我们暂且称他们为“绅士”。其实，如果我们稍加观察一下就可以看出，其中一位不配称“绅士”，他身材矮小，长相并无独特之处，但他却拥有着洋溢着自信、自尊和荣光的气质。他的衣服穿着有层次感，一件背心的绿色与那件蓝色的裤子相得益彰。他的右手拿着一只装满酒的酒杯，望着那杯酒，似乎在思考什么。他的右手拿着一只装满酒的酒杯，望

第一章 给读者介绍一位好心人

二月的某一天，天气依然比较寒冷。黄昏时分，在一场庆祝活动开展的餐厅里，两位绅士相对而坐，喝着酒。他们没有服务生在旁边侍候，他们夹着酒瓶，好像在商量什么重要的事情。

为了便适用于读者阅读，我们暂且称他们为“绅士”。其实，如果我们稍加观察一下就可以看出，其中一位不配称“绅士”，他身材矮小，长相并无独特之处，但他却拥有着洋溢着自信、自尊和荣光的气质。他的衣服穿着有层次感，一件背心的绿色与那件蓝色的裤子相得益彰。他的右手拿着一只装满酒的酒杯，望着那杯酒，似乎在思考什么。他的右手拿着一只装满酒的酒杯，望

Figure 2.3 Example of quasi-comparable websites

---

Chapter 1

In Which the Reader Is Introduced to a Man of Humanity

Late in the afternoon of a chilly day in February, two gentlemen were sitting alone over their wine, in a well-furnished dining parlor, in the town of P - - - , in Kentucky. There were no servants present, and the gentlemen, with chairs closely approach by to discuss some subject with great earnestness.

For convenience sake, we have said, thereto, two gentlemen. One of the parties, however, when critically examined, did not seem, strictly speaking, to come under the species. He was a short, thick-set man, with coarse, commonplace features, and that swaggering air of pretension which marks a low man who is trying to elbow his way upward in the world. He was much overdressed, a gray coat, the very color, a blue neckcloth, bedamped, grey with yellow spots, and matched with a41

intervals with various profane expressions, which not even the desire of being graphic in our account shall induce us to transcribe.

His companion, Mr. Shelby, had the appearance of a gentleman, and the arrangements of the house, and the general air of the housekeeping, indicated easy, and even upstart circumstances. As we before stated, the two were in the midst of an earnest conversation.

"That is the way I should arrange the matter," said Mr. Shelby.

"I can't make trade that way -- I positively can't, Mr. Shelby," said the other, holding up a glass of wine between his eye and the light.

---

Figure 2.3 Example of quasi-comparable websites


9
2.2 Approaches to Mine Parallel Documents

There are two categories of approaches to mine parallel documents from the web, the structure based methods ([6], [7], [8]) and the content based methods ([1], [2], [5], [3] [4], [9]). The structure based approaches mainly uses URL, HTML structure, hyperlinks to find pre-linked parallel documents on parallel websites whereas the content based approaches can find extra parallel resources from comparable and quasi-comparable websites by aligning the content.

2.2.1 Structure Based Approaches

Parallel websites have different language versions of the same content on the same website. Parallel documents are already aligned on parallel websites with one of the following features:

- Generalizable URL pattern (/en/, /zhs/...) [6]
- Anchor to each other with language identifier (English, 中文...) [6]
- HTML DOM Tree structure (same tree structure) [7]

Structure based approaches use the above characteristics to match parallel webpages with very high precision.

In previous works, [6] used (1) parent pages containing links to versions of one document in different languages and (2) sibling pages that contain links to the translation of the current documents. For each webpage on a parallel website, they try

---

to access the same page in another language by directly modifying the URL based on a pre-defined pattern. They also rely on the URL and anchor text to spot language specific version of the documents. If there is a hyper-link with the anchor text which indicates a language, such as "English", "Français" or "中文版 (Chinese Version)", the hyper-linked page will be considered as parallel document of the current one.

A structural alignment using the DOM tree representation was proposed by [7] to align parallel documents by using the HTML structure. They identify the translational equivalent texts and hyperlinks between two parallel DOM trees to find parallel documents.

However, the web is a heterogeneous collection of documents that extend far beyond bilingual and comparable pages with obvious structural features, such as similar URLs or common titles. Structural features only work for bilingual websites or document pairs that are already linked by editors. As only a small portion of websites are in the parallel website category, we cannot use this approach to crawl parallel resources from the entire web. It is also hard to discover new parallel websites automatically.

2.2.2 Content Based Approaches

Content based approaches typically align parallel documents by the contents. Those methods in previous works usually required to create a local archive of all documents in both source and target languages.

[2] and [3] proposed to download all source language and target language documents and then perform Cross Language Information Retrieval (CLIR) to extract candidate parallel documents.

![Diagram](https://via.placeholder.com/150)

(1) Download all documents in both languages

(2) Alignment on documents in local archive

Figure 2.4 Typical content based approach used in previous works

The steps of these approaches can be generalized and shown in Figure 2.4. All of them require a large local archive of both source and target documents. This can be very costly when we have to query the entire web.

The local index has to be updated regularly as the World Wide Web is updating continuously. Building and maintaining the local archive of trillions of documents is extremely costly in terms of CPU consumption, network bandwidth usage as well as disk storage utilization. Parallel document alignment algorithm cannot start until all documents in both languages have been downloaded.
Previous content based methods still require downloading all webpages in target languages even if we only want a small portion of source documents under a specific domain to be aligned.

Moreover, [5] uses statistical machine translation (SMT) system to translate all documents into target language to build a query index. Due to the complexity of machine translation algorithms (typically \(O(n^3)\)), it is waste of resource to download all target language documents, machine translate them, then select the desired candidate parallel documents.

Steps used in finding target documents can be replaced by using search engine APIs, such as Google or Bing, with search query generated from source documents to save CPU and bandwidth consumption in downloading trillions of target documents into a local achieve.

Figure 2.5 Basic steps of content based approach using search engine APIs.

We use the online mode which processes one source document at a time. Most research institutions interested in mining parallel documents do not possess a large number of CPUs or storage on the scale of the world’s top search companies. It is also desirable that any site can scale the mining speed and volume according to the computing resources available to them. There is no need for research institutions to crawl all target documents and build a local search engine which saves a lot of computational resources as well as bandwidth.
Changing of ranking algorithm is a new challenge to the approaches that query the web using search engines APIs. We do not want to refine the algorithm frequently in long term mining. The algorithm should communicate with the search engine interactively to adapt a new algorithm. The search query relevance score (SQRS) we proposed is used to evaluate the quality of each search query and monitor the search engine feedback during the search.

To this end, we propose a new low bandwidth and high precision content based method to on the one hand complement structural matching, and on the other hand reduce the complexity of content matching.

[11] proposed a mining approach on selected Chinese news article containing cue phrases, such as “根据外電報道” (According to foreign media). They proposed multiple algorithms to locally rank keywords which are generated from the source documents. Then they search the ranked keyword set in search engine API and find parallel documents. In non-oracle queries, which keywords are extracted from source documents, a maximum 40% of the documents found were parallel while the remaining pairs were comparable. This is a benchmark in mining precision using search engine APIs.

As the parallel resources mined are often used to improve SMT systems or yield bilingual lexicons, it is desirable that the mining output is of high precision.
CHAPTER 3

Our Content Based Approach Supplemented with Structural Information

Our proposed content based approach (Figure 3.1) primarily aims to discover parallel documents from all kinds of parallel, comparable or quasi-comparable websites on the World Wide Web. We take advantage of online search engines to find candidate documents but against the search result, thereby saving bandwidth and cost and avoiding the need to crawl all target documents and for storage in an archive.

Content based approach queries the document in target language using keywords from documents in the source language. In our approach, queries are generated from source documents and expanded dynamically and interactively by search result quality score (SQRS) as feedback. Our algorithm can automatically adapt to the change of ranking algorithms.

As neither machine translation of the full text nor pre-download of target documents is needed for our approach, mining can start instantly when a batch of source documents from the same domain has been downloaded.
We mainly focus on the precision of output parallel documents. The query expansion feedback score is the key in improving the precision of target documents found. We also use DTW and $R^2$ regression as verification methods to further ensure the quality of parallel document pairs. Although the target documents in search result always have some relationship with the source document in common keywords, but they may not in the sense of comparable or parallel. We do not allow non-comparable document pairs in the output. If a source document is found to have no translation in the target language or the target document fails the verification, the system simply returns <not-found>.

### 3.1 Source Document Representation

Online search engines use keywords to find documents. We cannot enter source documents with thousands of words directly into a search engine. We have to convert
text into keywords to perform automated queries. Each single keyword may exist in multiple articles. However, they can identically represent a document if several keywords are put together as a keyword set [10]. To find document in target language, we translate each keyword into target language using bi-lexicon to form the initial query.

A query does not always yield the desired target document by directly using all the translated keywords from source document. The reasons are:

- Keyword translation might not correspond to the actual words in the target document
- Certain keywords in the target documents might have been removed by content editors
- There are errors in keyword translation or selection
- Number of keywords exceeds search engine limit

It is essential to select appropriate keywords and use correct translation to find the desired target document in a search engine. Two conditions that an appropriate keyword set should satisfy are: (1) they should represent the document exclusively [10] (2) they should have unique or common translation in both languages.

We suggest that words with high Term Frequency and Inverse Document Frequency (TF-IDFs) and English words in Chinese text are usually keywords that fulfill both conditions above.

\[ K = K_T \cup K_E \]

\( K_T \): set of words with high TF-IDF score

\( K_E \): set of English words in Chinese documents
To obtain TF-IDFs that are representative of the keyword in the source document, in our experiments, they are trained from all source documents under the same domain name (e.g. www.ftchinese.com). This can help us avoid words that exist in navigation area (banner, footer) and advertisements since they have very low IDF in the website. Sometimes, source documents may not be downloaded from the same website. Domain-wise TF-IDF training is not yet possible. In such cases, it is acceptable to use TF-IDF trained in the general domain.

Keywords in $K_E$ are more important because most of them are words actually used in the target document. There is no translation error since words in $K_E$ are in the target language already. However, in many cases, there are additional words in $K_E$ so that we cannot find any document by directly searching $K_E$. The effect of additional wrong keywords to search result will be discussed in the next section. Our method combines $K_E$ with $K_T$ to obtain the keyword set and use words in $K_E$ first.

### 3.2 Search Target Documents with Search Query Relevance Score (SQRS)

Search engine takes multiple criteria, including keyword significance, domain popularity, date, popularity, page rank and etc., to return the most relevant documents that match the query. Each keyword may have more than one possible translation in the b lexicon. We do not know the words that are actually used in the target documents when we are generating keywords from source document.

In previous studies, [11] used local ranked keywords and query each document once. However, search engine ranking algorithm and index are changing over time. The same query may not always yield the desired document in the search result. To maximize the
possibility of finding the target document, we take search engine feedback into account
to interactively amend queries.

We proposed to use SQRSs which evaluate the relevance of the search result and the
search query. Then we amend the keywords according to the SQRS feedback. By using
SQRS as feedback, we can:

- Choose better keywords set
- Use the correct translation of keywords
- Adopt change of search engine ranking algorithm

Commercial search engines omit some keywords when there is no document in the
index containing all keywords. In such cases, the rank of documents usually changes
significantly.

The following example shows the ranking change caused by erroneously translated
keyword. The search result of two search queries are generated from the Chinese
version of My Space launching new version of website7. “|” indicates separator of
keywords.

Query 1:

    myspace | mike jones | facebook | san francisco | new | website

Query 2:

    myspace | mike jones | facebook | san francisco | new | website | fashion

Figure 3.2 Search result of Query 1 (a) and 2 (b) on Google.com
In Query 1, the oracle target document was the topmost in the search result. The short summary contains every keyword we entered in the query. The SQRS generally decreases with the rank of webpages. (For definition and calculation of SQRS, see Equation 3.1)

<table>
<thead>
<tr>
<th>SQRS</th>
<th>Search Engine Omitted Keyword</th>
<th>Rank in Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.742</td>
<td>-</td>
<td>1 (Target)</td>
</tr>
<tr>
<td>5.174</td>
<td>web(site)</td>
<td>3</td>
</tr>
<tr>
<td>4.951</td>
<td>web(site)</td>
<td>2</td>
</tr>
<tr>
<td>4.663</td>
<td>web(site)</td>
<td>4</td>
</tr>
<tr>
<td>4.545</td>
<td>web(site)</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 3.1 SQRS of Query 1

In Query 2, we added fashion which is the English translation of “新潮” (but the actual English version used hottest). The rank of search result changed and each summary omitted at least one keyword in the query. The SQRS does not align with the search result yet.

<table>
<thead>
<tr>
<th>SQRS</th>
<th>Search Engine Omitted Keyword(s)</th>
<th>Rank in Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.155</td>
<td>fashion</td>
<td>5 (Target)</td>
</tr>
<tr>
<td>3.951</td>
<td>web(site)</td>
<td>fashion</td>
</tr>
<tr>
<td>5.867</td>
<td>website</td>
<td>3</td>
</tr>
<tr>
<td>0.871</td>
<td>mike</td>
<td>new</td>
</tr>
<tr>
<td>-2.921</td>
<td>mike jones</td>
<td>new</td>
</tr>
</tbody>
</table>

Table 3.2 SQRS of Query 2
This phenomenon implies that the document containing all keywords in Query 2 does not exist on the web. The recently added keyword *fashion* must be erroneously translated.

In many similar cases, an erroneously translated keyword can pollute the query quality and decrease the rank of the target document. Parallel document mining cannot rely on the document rank of a search engine. The system must have a mechanism to detect the problem when expanding the query. Otherwise, a batch of irrelevant documents will be downloaded which need to be filtered out.

To avoid adding erroneously translated keywords to further reduce the amount of undesirable documents downloaded, we introduced the *search query relevance score (SQRS)* that describes how well the search result is and how we can refine the query if the desired document is not presented.

The SQRS score is determined by comparing the query with highlighted keywords in the search result. Generally, a webpage has higher SQRS if the summary contains more keywords that match the query. (Equation 3.1)

We ran experiments on 112 randomly selected source documents and observed the SQRSs in the search result. As shown in Table 3.3, 81 (72.3%) target documents have the highest SQRS among other URLs in the search results. It implies the SQRS are an effective measure of query formation and keyword translation.

<table>
<thead>
<tr>
<th>Source Documents</th>
<th>Target Documents Have Largest SQRSs</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>112</td>
<td>81</td>
<td>72.3</td>
</tr>
</tbody>
</table>

Table 3.3 Result quality and SQRS

22
\[ Q = (k_1, k_2, \ldots, k_v) \]

\[ k_i = (w_1, w_2, \ldots, w_n) \]

\[ \Gamma K_i = \{ (w_a, \ldots, w_b) | 0 < a < b \leq J \} \]

\[ \text{count}(c, t) = \# \text{ of occurrence of } c \text{ in } t \]

where \( \delta(c, t) = \begin{cases} 0 & \text{count}(c, t) > 0 \\ 1 & \text{count}(c, t) = 0 \end{cases} \)

\[ SQRS(Q, T) = \sum_{c=1}^{n} \left( \sum_{s \in \Gamma K_s} \left[ \log(\text{count}(s, T) + 1) \right] - \delta(k_c, T) \right) \]

where \( Q \) is the query, \( k \) is the keyword, \( w \) is the English word and \( T \) is the short text with highlighted keywords in the search result.

Equation 3.1 Definition of SQRS

Although the query may include multiple translations of a keyword in bilingual lexicon dictionary, the SQRS ensures that there is minimum adverse effect from incorrect translations and irrelevant keywords.

SQRS can avoid erroneously translated keywords on the one hand; on the other hand, it can also help the system to select keywords because some keywords extracted from source document may not appear in the target document. The negative effect of searching irrelevant keywords is similar to adding erroneously translated keywords.

There are many permutations of keyword sets for each document. It is unlikely to use the correct keyword set and get the desired document in the first search attempt. To
improve the precision of the keyword set, we use SQRS as relevance feedback as shown in Figure 3.3.

![Flowchart of query expansion algorithm](image)

**Figure 3.3 Flowchart of query expansion algorithm**

First, we rank the keywords in $K_T$ by the TF-IDF scores. Next, the query is expanded by SQRS. When keyword $w$ is added to the current query, we compare the maximum $SQRS$ among top $n$ results with the previous highest score $SQRS$, without $w$. $w$ will be discarded from the keywords if $SQRS_w < SQRS$, or simply caused an empty search result. Otherwise, the query will be expanded by adding $w$.

<table>
<thead>
<tr>
<th>Source Doc</th>
<th>Doc Pairs</th>
<th>Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without SQRS</td>
<td>1000</td>
<td>217</td>
</tr>
<tr>
<td>With SQRS</td>
<td>1000</td>
<td>243 (+12.0%)</td>
</tr>
</tbody>
</table>
Table 3.4 Comparison of target document found with and without SQRS

The search engine returns the total number of target documents for each query. If this number is less than a threshold \( M \), we verify the parallelity of top-ranked documents in the next step and stop the query expansion. To save network bandwidth, the system only considers \( K_{Max} \) words with the highest TF-IDF scores and directly output <not-found> if no target document is found.

### 3.3 Document Verification

Documents pairs found in the search engine are candidates of parallel documents. Even though we have SQRS to measure the quality of keywords translation and selection, however, it is based on keywords only. We need to have a step to verify the parallelness of document pairs to guarantee the precision.

All candidate document pairs are subjected to pass the parallelness check before output. Output is <not-found> if all candidate target documents failed the verification process. We propose using both dynamic time warping (DTW) and \( R^2 \) regression as in our previous work [12] on every pair of the source and targets document to evaluate the parallelness and discard those pairs that failed the verification.

#### 3.3.1 Dynamic Time Warping (DTW) Score

\[
DTW(i_m, i_n) = c + \min \begin{bmatrix}
DTW(i_m, i_n) & \ldots & DTW(i_m - d, i_n) \\
\vdots & \ddots & \vdots \\
DTW(i_m, i_n - d) & \ldots & DTW(i_m - d, i_n - d)
\end{bmatrix}
\]

Equation 3.2 DTW with local distance of \( d \)
DTW alignment is much faster than machine translation (MT). We measure the word level DTW score between source document and target document with local constrain of \(d\) (Equation 3.2). Stop words are removed from the English text before DTW processing.

We use a bilingual lexicon (dictionary) to calculate the DTW path of each pair. If there is an entry in the bi-lexicon for \(i^{th}\) word of source document and \(j^{th}\) word of the target document (i.e. Chinese and English) respectively, the score of point \((i,j)\) is 1, otherwise 0. The total score is normalized by maximum number of steps (moves) from \((0,0)\) to \((m,n)\) to convert DTW score to a number between 0 and 1.

Parallel document pairs tend to have a path close to the diagonal line with high DTW score. Figure 3.4 shows the DTW paths and scores of a parallel document pair and a non-parallel pair. The precision of output sentences increases if the DTW score threshold is set higher. The non-parallel documents also have overlapped words but the word sequences are too different.

Table 3.5 is the relationship between DTW scores and the precision of candidate pairs based on our experiment and human evaluation of pairs.

Figure 3.4 DTW of parallel and non-parallel document pairs
<table>
<thead>
<tr>
<th>DTW</th>
<th># Pairs</th>
<th># Parallel</th>
<th>Precision %</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0.45</td>
<td>122</td>
<td>121</td>
<td>99.18</td>
</tr>
<tr>
<td>&gt;0.40</td>
<td>224</td>
<td>219</td>
<td>97.77</td>
</tr>
<tr>
<td>&gt;0.35</td>
<td>298</td>
<td>288</td>
<td>96.64</td>
</tr>
<tr>
<td>&gt;0.30</td>
<td>354</td>
<td>337</td>
<td>95.20</td>
</tr>
<tr>
<td>&gt;0.28</td>
<td>389</td>
<td>364</td>
<td>93.57</td>
</tr>
<tr>
<td>&gt;0.26</td>
<td>429</td>
<td>389</td>
<td>90.68</td>
</tr>
<tr>
<td>&gt;0.25</td>
<td>456</td>
<td>403</td>
<td>88.38</td>
</tr>
<tr>
<td>&gt;0.24</td>
<td>488</td>
<td>415</td>
<td>85.04</td>
</tr>
<tr>
<td>&gt;0.22</td>
<td>545</td>
<td>417</td>
<td>76.51</td>
</tr>
<tr>
<td>&gt;0.20</td>
<td>627</td>
<td>426</td>
<td>67.94</td>
</tr>
</tbody>
</table>

Table 3.5 DTW and precision of candidate pairs

The output pairs are mainly comparable documents of the same topic if the DTW score are upper limited to a threshold.

3.3.2 $R^2$ Regression

The parallel documents contain parallel sentences that may have different word orders, especially in the case of English and Chinese. The DTW score may be affected by different word order. We propose to use $R^2$ regression as an additional feature.

Score measures the deviation of the matching path of shared words in both documents from the diagonal. Stop words are not included in the bi-lexicon dictionary.
The size of source and target documents is usually different. The $R^2$ score is normalized by the size ratio to make it comparable among document pairs.

$$R^2 = R^2_{score} / sizeof(T_{src}) * sizeof(T_{tar})$$

(a) $R^2$ of a parallel pair ($R^2=1.3E^4$)

(b) $R^2$ of a non-parallel pair ($R^2=1.2E^6$)

Figure 3.5 $R^2$ of parallel and non-parallel document pairs

3.3.3 Combining DTW and $R^2$

DTW score helps filter out non-parallel pairs and $R^2$ is introduced as a supplementary feature to further improve the precision of extracted parallel documents.

A comparison of using these measures is shown in Table 3.6.

<table>
<thead>
<tr>
<th></th>
<th>DTW (&gt;0.22)</th>
<th>$R^2$ (1.0E-5,1)</th>
<th>DTW+$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td># Pairs</td>
<td>545</td>
<td>534</td>
<td>481</td>
</tr>
<tr>
<td># Parallel</td>
<td>417</td>
<td>403</td>
<td>399</td>
</tr>
</tbody>
</table>


| Precision % | 76.51 | 75.47 | 82.95 |

Table 3.6 Mining precision of DTW and R²

### 3.3.4 Structural Features

The final step of verification uses structural features of the document pair candidates:

- **Language**: mined document should be in the target language
- **Absolute size**: neither document should be too small/large in file length
- **Size difference**: two documents must be of similar size
- **Document type**: both documents must be content page in a website

### 3.4 Improve Recall by Structure Based Information

Since search engines rank target documents by various criteria, including page rank, some bilingual website documents might be missed by the pure content based approach. We propose to supplement our approach with the URL matching patterns or anchors if the content based method has found several pairs of source and target documents having the same pattern.

We examine the pairs found by the content based method and look for any characteristics that match the structure based patterns, including URL, anchors and etc. Then, we apply the patterns to all source web pages under the same hostname. This will greatly increase the amount of parallel documents in the output as the recall of content based method is usually lower than the structure base methods.
The experiment result shown in Table 3.7 illustrates the number of target documents found by content based method and URL matching for the same set of source documents on parallel websites. URL matching generally improves the recall by 2 times.

<table>
<thead>
<tr>
<th>Source</th>
<th># of Doc</th>
<th>Content Based</th>
<th>Content Based + URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTChinese</td>
<td>11,009</td>
<td>2,968</td>
<td>9,066</td>
</tr>
<tr>
<td>WSJ (CN)</td>
<td>3,327</td>
<td>1,002</td>
<td>3,120</td>
</tr>
</tbody>
</table>

Table 3.7 Comparison of content based method and URL matching

All pairs found by both content based and structure based methods are also subjected to passing the verification process introduced in Section 3.3.

<table>
<thead>
<tr>
<th>Source</th>
<th># Chinese Docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ftchinese.com</td>
<td>11,009</td>
</tr>
<tr>
<td>cn.wsj.com</td>
<td>3,327</td>
</tr>
<tr>
<td>cn.reuters.com</td>
<td>8,570</td>
</tr>
<tr>
<td>forbeschina.com</td>
<td>6,281</td>
</tr>
<tr>
<td>fortunechina.com</td>
<td>593</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>29,780</strong></td>
</tr>
</tbody>
</table>

Table 3.8 Source documents for pure content based approach
CHAPTER 4
EXPERIMENT SETUP AND RESULT

4.1 Experiment setup

We evaluate our approach on two sets of experiments. They are conducted on a single machine with 8 processing cores. Source and target documents in the experiments are Chinese and English respectively.

4.1.1 Website Parsing

To compare the DTW and $R^2$ of each document pairs, we have extracted the main content of the web page. We find the main content of the document using ExtMainText\(^8\) in python.

However, ExtMainText computes the text density on a HTML page, which defined as:

$$\rho_{text} = \frac{l(Text)}{l(Text) + l(Tag)}$$

A specified threshold of $\rho_{text}$ is required for different sites or even different documents to get the main content extracted. We also need to adjust the value for different languages.

We have implemented an algorithm to select $\rho_{text}$ automatically.

\(^8\) http://www.elias.cn/En/ExtMainText/
We calculated $\rho_{text}$ for each line. Then classified $\rho_{text}$ from the first line to the last, i.e. lines with similar $\rho_{text}$ will be grouped into the same class. The class including most number of lines is extracted as the main text of the page.

In our experiment, we use different TF-IDF for each domain name. For example, we train TF-IDF for ftchinese.com using all source documents under this domain name and this TF-IDF which are used for these source documents only.

Words appearing in the header and footer will get very high in both TF and IDF and will be low weighted in the keyword selection.

We use the Chinese-English Translation Lexicon Version 3.0 from LDC\(^9\). The system can directly load the bi-lexicon in the same format for other language pairs.

### 4.2 Content Based Baseline

As a baseline of content based method, we directly enter English words in the original Chinese document in the search engine (Experiment i). Then, we add keywords ranked by TF-IDF to query the target document but do not perform SQRS to expand the query (Experiment ii).

Finally, SQRS is used to refine each keyword to get better results (Experiment iii).

Our approach is not search engine dependent. All experiments use both Google Search API and Bing Search API to search the keyword sets. Results from different search engines are merged together by a unique URL.

---

\(^9\) LDC Catalog Number: LDC2002L27
We generalize URL patterns (if any) from document pairs when we find some document pairs by the content based method on parallel websites. By structure based information, we extract more parallel webpages from parallel websites that follow those URL patterns.

4.3 Experiment 1: Find Target Documents

Source (Chinese) documents in our experiments are news from the following 5 agencies:

Parallel websites:

• (1) Financial Times Chinese (ftchinese.com)
• (2) Wall Street Journal Chinese (cn.wsj.com)

Parallel website contain both Chinese and English documents under the same host and can be aligned with URL matching.

Comparable/quasi-comparable websites:

• (3) Reuters China (cn.reuters.com)
• (4) Forbes China (forbeschina.com)
• (5) Fortune China (fortunechina.com)

Documents on quasi-comparable or comparable websites may have target documents on either the corresponding agencies’ global site (e.g. cn.reuters.com and www.reuters.com) or somewhere else. No parallel documents from such websites can be found by URL matching or other structure based methods.
To evaluate the precision of our content based approach, we applied our method to the above sites for target documents. The percentage of parallel documents that we can successfully find is highly dependent on the type of documents and search engine index. Calculating recall, on the other hand, is only possible for sites we already knew. For comparable or quasi-comparable sites, it is not possible to evaluate recall of the algorithm because:

- Some source documents may not have translation in the target language
- Target language pages may not be indexed by search engines
- Manual evaluation of all documents for recall calculation is not feasible

Thus, we mainly focus on the precision of the output pairs.

In the verification process, we discard the document pairs if:

- DTW score < 0.25 (Output document pairs have 88% precision)
- $R^2$ score < 1.0E-5
- Article size is too small (Only one or two sentences)
- Size of source and target are too different
- URL is root (/) under hostname (We consider content pages only)
- Text in wrong language

We manually evaluate the effectiveness of our method on randomly selected document pairs from the output of our content based method. Only strict parallel document pairs are considered as correct. The target precision of parallel documents in this experiment is 88%.

The result of Experiment 1 is shown as follows:
We directly searched all English keywords in Chinese documents and found 153 target documents (baseline). Then we search translation of top ranked TF-IDF keywords (ii). Query expansion with SQRS further improved by 23.56% of the output sentences compared to the baseline (Table 4.1).

<table>
<thead>
<tr>
<th>Source Doc</th>
<th>Doc Pairs</th>
<th>Sentences</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>1000</td>
<td>153</td>
<td>2483</td>
</tr>
<tr>
<td>ii</td>
<td>1000</td>
<td>217</td>
<td>2906</td>
</tr>
<tr>
<td>iii</td>
<td>1000</td>
<td>243</td>
<td>3068</td>
</tr>
</tbody>
</table>

i. Direct Search of $K_F$

ii. Top ranked keywords without SQRS

iii. Query expansion with SQRS

Table 4.1 Comparison of different methods

We also applied our content based method on larger scale source documents. Among the 29,680 Chinese documents retrieved from the five news agencies, we obtained 7,253 parallel document pairs with 88% precision by content based approach alone.

In many such cases, parallel document pairs are on different websites could be found neither by URL matching nor by content-based methods that used times stamps for matching.

With structure based information, we increase the output of the parallel documents from parallel websites. Table 4.2 shows that the URL matching can improve the output quantity a lot, compensating for the missing target documents with low page ranks.
For parallel bilingual websites, the content based method can find about 1/3 (33.3%) of the target documents compared to the content based method URL matching. It shows that, however, our query expansion with relevance feedback approach has higher recall than the 18% produced by the local ranked keywords in [11].

<table>
<thead>
<tr>
<th>Source</th>
<th># of Doc</th>
<th>Content Based</th>
<th>Content Based + URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTChinese</td>
<td>11,009</td>
<td>2,968</td>
<td>9,066</td>
</tr>
<tr>
<td>WSJ</td>
<td>3,327</td>
<td>1,002</td>
<td>3,120</td>
</tr>
<tr>
<td>Reuters</td>
<td>8,570</td>
<td>1,911</td>
<td>1,911</td>
</tr>
<tr>
<td>Forbos</td>
<td>6,281</td>
<td>1,166</td>
<td>1,166</td>
</tr>
<tr>
<td>Fortune</td>
<td>593</td>
<td>206</td>
<td>206</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>29,780</strong></td>
<td><strong>7,253</strong></td>
<td><strong>15,469</strong></td>
</tr>
</tbody>
</table>

Table 4.2 Output document pairs of Experiment 1

### 4.4 Experiment 2: Parallel Sentence Extraction for SMT

In order to obtain a sentence alignment for pairs of document, we first need to extract the proper content of each page and remove the header and footers that are of little interest and are unlikely to be parallel anyway.

For parallel documents, we can use additional features in parallel text to further improve effectiveness of the web parsing algorithm to extract the main content. We first segment the documents in sentences and filter out improper ones, such as English sentences containing Chinese characters, or Chinese sentence containing roman characters only. We then use DTW again to find a continuous path in the documents.
and extract the longest path. The header and footer will generally not align and will be discarded; only the chunk of true alignable content will be preserved.

Using this method, we manage to find the beginning and the end of source and target content and extract it. Then discarding pairs of document whose number of extracted sentences are too different, sentence alignment is performed on the remaining documents using the Champollion ToolKit, [8] which is already trained for Chinese-English document pairs.

Finally, we filter all the sentences using a simple word overlap score. Sentences whose lengths are too different or whose word overlap score is too low are discarded, to ensure a high precision at the end.

Among the 15,469 Chinese-English document pairs found in Table 4.2, we extracted 225,374 parallel sentence pairs with mining precision of over 97%. We evaluate the quality of those sentences for training machine translation with the Moses SMT engine.

We use Bilingual Evaluation Understudy (BLEU) [13] score to evaluate the quality of parallel sentences. BLEU score is given by comparing the machine translated text with professional human translated text. The closer the result, the higher the quality of the translation is. We compare the BLEU score obtained with a 4,097,357 sentence pairs corpus, manually aligned (baseline) and the BLEU score obtained with the same corpus, replacing 225,374 sentence pairs by the ones we extracted. The results are presented in Table 4.3. They are evaluated using the NIST MT06 evaluation set.

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>29.54</td>
</tr>
<tr>
<td>Our mined data</td>
<td>29.88</td>
</tr>
</tbody>
</table>
Table 4.3 BLEU score obtained for SMT (Experiment 2)

These results show that our set of sentences, together with a larger parallel corpus, yield results similar to the one obtained with manually aligned sentences only.

4.5 Experiment 3: Bilingual Lexicon Extraction

The extracted sentences from 15,469 documents have been processed for rare word translation extraction [13]. Jaccard similarity is used to measure words association in documents of comparable corpus. Log-likelihood and cosine similarity are used in the extraction process to normalize and compare the context vector respectively.

As a result of the experiment using document pairs mined by our system, 80% of F-Measure is yielded for Chinese-English rare word lexicon extraction. This result is comparable to the F-Measure of another language pair of French English using aligned document pairs from Wikipedia.

4.6 Experiment 4: Parallel Document Mining for Hindi-English

We have also expanded our tool to mine parallel documents of Hindi-English pairs. The source documents are in Hindi and the target documents are in English. We changed the lexicon of Chinese-English to Hindi-English. All the other setup is the same as that used in Experiment 1.

The preliminary result showed the tool works on Hindi-English parallel document extraction. For Hindi-English pairs, we have yielded the precision of 86.4% with DTW>0.25 on 187 human evaluated documents pairs. See Table 4.4 for details.
The precision of Hindi-English pairs are slightly lower than the Chinese-English pairs. It is affected by the size and quality of bilingual lexicon.

The recall of the experiment is about 20%. However, the recall does not reflect the performance of the tool because it depends on search engine index coverage and types of source document.

<table>
<thead>
<tr>
<th>DTW</th>
<th>Source documents</th>
<th>Correct</th>
<th>Precision</th>
<th>Precision for zh-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0.25</td>
<td>22</td>
<td>19</td>
<td>86.4%</td>
<td>88.4%</td>
</tr>
<tr>
<td>&gt;0.24</td>
<td>24</td>
<td>20</td>
<td>83.3%</td>
<td>85.0%</td>
</tr>
<tr>
<td>&gt;0.22</td>
<td>33</td>
<td>24</td>
<td>72.7%</td>
<td>76.5%</td>
</tr>
<tr>
<td>&gt;0.20</td>
<td>59</td>
<td>34</td>
<td>57.6%</td>
<td>67.9%</td>
</tr>
</tbody>
</table>

Table 4.4 Result of Experiment 4

4.7 System Performance and Scalability

We carried out our mining experiments on the workstation with 8 state-of-art CPU cores. The average time taken for each source document is 30 seconds which is only bottle-necked by the limitation of search engine APIs.

As the TF/IDF scores are pre-trained only from the source documents and our content based approach mines the target document for each source document individually, the system can be easily scaled to run in parallel on multiple servers.

File storage can be solved by using distributed file system or storage servers to enable large scale mining from the entire web.
The database manages the state of each URL and outputted document pairs. Mainstream database management systems such as MySQL, SQL Server and Oracle can also be scaled using existing solutions.

4.8 Future Works and Discussion

In this thesis, we have shown that the system works on finding English target documents by using source document in Chinese or Hindi to mine Chinese-English or Hindi-English parallel documents. The mining precisions are similar for these two language pairs. We only extract parallel sentences from aligned Chinese-English parallel documents and evaluate the result by using the machine translation system and BLEU scores. We will also implement a sentence extractor for the Hindi-English sentences and evaluate the quality of those parallel sentences.

In future works, we may extent the system to other languages, for both source and target documents. For some language pairs, the quantity of parallel resources available is very limited. The mining of parallel documents in those language pairs will have significant contribution to the size of parallel corpus and improve the quality of machine translation of the two languages.

It is noticeable that the recall of our content based approach is not as high as the structure based approach. For comparable and quasi-comparable websites, it may miss some parallel documents that can only be found by the content based method. We may improve the recall and further increase the precision by:

(a) Increase bilingual lexicon / dictionary size
The dictionary we used is directly downloaded from the LDC which is small compared to the words in documents we have processed. We can use a large dictionary to increase the accuracy of translation and DTW verification.

(b) Use other keyword extraction methods

We use keywords in target language \(K_T\) and keywords with highest TF-IDF \(K_T\) to form queries and search for target documents. Other keyword sets such as name entities and temporal information from the source documents also can be keywords in this step. They may need to have additional training for the source document domain in order to be used by the system.

(c) Add other search engines

Google and Bing are used in the system for finding target documents. However, these two largest commercial search engines rank webpages mainly rely on criteria other than keywords relevance. This may cause difficulty in finding target document which is not new. We may integrate other search engines to search target documents and rank by keywords relevance only.

There is a usage limitation for search engine APIs of Google and Bing which is the bottleneck of our mining system. The speed of the system in parallel documents mining can also be improved if there is another search engine which has looser usage limitation for its API.
CHAPTER 5
CONCLUSION

In this thesis, we have proposed a content based CLIR approach which searches any part of the web to find parallel documents without the limitation of URL-matched bilingual websites.

We use search query relevance score (SQRS) to ensure translation correctness and measure relevance between keywords and search result to further ensuring the keywords we use represent the source document. Using a supplementary verification process, the web documents are then filtered by dynamic time warping and regression scores.

Experimental results show an 88% mining precision on the parallel documents extracted from parallel, comparable and quasi-comparable web sites.

Another experiment on extracting bilingual sentences from the mined documents shows that the sentence extraction adds another layer of verification which further improves precision from 88% to 97%.

SMT experiments on using our mined parallel sentences, together with a larger baseline training set, to train an SMT system show comparable performances from using our data to that of using manually aligned bilingual sentences. BLEU score is improved from 29.54 to 29.88.

Moreover, the precision of parallel resources mined can be adjusted by users to balance between quality and quantity of document pairs. It is also possible to mine comparable documents by limiting the DTW and regression score to a certain range.
The parallel documents mined by our approach are also used as a part of the corpus to extract translations of rare words. 80% F-Measure are obtained for the extraction of rare Chinese-English lexicon.

Our proposed method does not require full-text machine translation, nor does it require downloading all documents in the target language into an archive for document matching, thereby saving lots of network bandwidth and computational resources. In addition, Mining process can start once a batch of source documents is downloaded and the TF-IDFs are trained using the source documents on the same domain name.

The new content based approach works on the heterogeneous web. We use Google and Bing APIs to automatically search parallel documents by keywords with SQRS for search result feedback.

The experiment result shows the supplementary structured base method of URL matching mined parallel documents from parallel websites to be twice as many as the pure content based method. Our system works on multiple language pairs in parallel documents mining and it is extendable to more language pairs. We ran experiment to mine Hindi-English parallel documents and the mining precision is similar to that we get in Chinese-English pairs.

Our system is scalable to run on multiple servers simultaneously and is linear in time to the number of input source documents. It can also be run continuously to discover and mine for newly added web documents that were not there previously. It is also extendable to mine for parallel documents in multiple target languages at the same time.
REFERENCES


European Chapter of the Association for Computational Linguistics (EACL’06), 2006, pp. 16-23.


APPENDIX A
SYSTEM DOCUMENTATION

1. System Requirement

- Java Run Environment (JRE) 1.5 or later
- MySQL Database 5.0 or later
- Python 2.6 or later

2. Package Content

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSM_FindParallel.jar</td>
<td>Parallel sentence miner, to search parallel target documents for given source documents in the database.</td>
</tr>
<tr>
<td>PSM_Output.jar</td>
<td>Final verification of parallel pairs under specific DTW threshold. Copy to output directory and perform sentence alignment.</td>
</tr>
<tr>
<td>PSM_ZHIDF.jar</td>
<td>TF-IDF trainer for each domain. Please run this jar before finding parallel documents</td>
</tr>
<tr>
<td>File Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PSM_Spider.jar</td>
<td>Web spider for source documents. It will crawl the web and perform regular updates for URLs in the database.</td>
</tr>
<tr>
<td>PSM_WebCrawler.jar</td>
<td>Webpage download scheduler. It will download/update each URL in the database.</td>
</tr>
<tr>
<td></td>
<td>There is no need to run this file directly. Spider will call this file.</td>
</tr>
<tr>
<td>PSM_AnchorAnalyzer.jar</td>
<td>Analyze hyper-links in each webpage downloaded. New URLs will be added to the database.</td>
</tr>
<tr>
<td></td>
<td>There is no need to run this file directly. Spider will call this file.</td>
</tr>
<tr>
<td>s1-wget.jar</td>
<td>This is a wget implementation in Java. No need to call this file directly.</td>
</tr>
<tr>
<td>segment-zh.sh</td>
<td>This is for Chinese text segmentation. No need to call this file directly.</td>
</tr>
<tr>
<td>File Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ExtMainText.py</td>
<td>A python script for extracting main text in web page. Used by TF-IDF trainer and Parallel Finder.</td>
</tr>
<tr>
<td>ldc_cedict.txt</td>
<td>Bi-lexicon file. Default is Chinese-English from LDC. User may change this file for other language pairs. Format (1 entry per line): Lang1&lt;tab&gt;/Lang2-1/Lang2-2/.../Lang2-N/</td>
</tr>
<tr>
<td>cfile.txt</td>
<td>Sample configure file. See installation section for details. Need to change</td>
</tr>
<tr>
<td>mysql-connector-java-5.1.10-bin.jar</td>
<td>Java library for MySQL database.</td>
</tr>
<tr>
<td>json.jar</td>
<td>Java library for JSON.</td>
</tr>
<tr>
<td>database.sql</td>
<td>Database structure. Please import to MySQL server.</td>
</tr>
</tbody>
</table>
3. Installation

1. Install MySQL Server and import `database.sql` to create tables required by the tool. Make sure a new schema has been created and the user has enough privilege to perform the following operation on the schema: `SELECT`, `UPDATE`, `INSERT` and `DELETE`.

2. Unzip and put all files under the same directory and modify the configuration file (sample configuration file: cfile.txt).

<table>
<thead>
<tr>
<th>Key</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>storepath</td>
<td>Path to store data files</td>
</tr>
<tr>
<td>dbserver</td>
<td>Database server name of IP</td>
</tr>
<tr>
<td>dbname</td>
<td>Database name</td>
</tr>
<tr>
<td>dbuser</td>
<td>Database user name</td>
</tr>
<tr>
<td>dbpwd</td>
<td>Database password</td>
</tr>
<tr>
<td>gsearchKey</td>
<td>Google Search API key</td>
</tr>
<tr>
<td>bsearchKey</td>
<td>Bing Search API key</td>
</tr>
<tr>
<td>lang1</td>
<td>Source language, two letters</td>
</tr>
<tr>
<td>lang2</td>
<td>Target language, two letters</td>
</tr>
<tr>
<td>threads</td>
<td>Number of threads allowed</td>
</tr>
<tr>
<td>rLimit</td>
<td>Number of search results (M). Default: 2000</td>
</tr>
<tr>
<td>dtwThreshold</td>
<td>DTW threshold. Default: 0.25</td>
</tr>
<tr>
<td>useBingTranslation</td>
<td>Use Bing or Google to translate keywords.</td>
</tr>
<tr>
<td>addKeywords</td>
<td>Additional keywords for each query. Default: &lt;Empty&gt;. May be used to specify website. E.g. <code>site:www.abc.com</code></td>
</tr>
</tbody>
</table>
4. Commands of the Tool

Crawl and Update Source Documents

```
java -Xbootclasspath/a:mysql-connector-java-5.1.10-bin.jar:json.jar -jar PSM_Spider.jar
<configuration file> <number of threads>
```

Train TF-IDF

```
java -Xbootclasspath/a:mysql-connector-java-5.1.10-bin.jar:json.jar -jar
PSM_TFIDF.jar <configuration file> <data directory> [<domain name>]
```

Domain name is optional. It will only train for webpages under the specific domain name if provided.

IDF dump files are under <data directory>/idfdumps/<domain name>_<lang>.txt

Format:

```
Word	DF
```

Parallel Documents Mining and Sentence Extraction

```
java -Xbootclasspath/a:mysql-connector-java-5.1.10-bin.jar:json.jar -jar
PSM_FindParallel.jar <configuration file>
```

Output documents and extract parallel sentences
5. Database Structure

This part is about the tables created by database.sql.

**Urls Table**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>txt_url</td>
<td>URL of the document</td>
<td>-</td>
</tr>
<tr>
<td>txt_domain</td>
<td>Domain name in the URL</td>
<td>-</td>
</tr>
<tr>
<td>id</td>
<td>Automatic Unique ID</td>
<td>Auto</td>
</tr>
<tr>
<td>time_add</td>
<td>Time when the record is added</td>
<td>&lt;null&gt;</td>
</tr>
<tr>
<td>time_lastaccess</td>
<td>Time when the record is accessed</td>
<td>&lt;null&gt;</td>
</tr>
<tr>
<td>int_error</td>
<td>Number of errors for processing the URL</td>
<td>0</td>
</tr>
<tr>
<td>bool_active</td>
<td>The document will be processed if True</td>
<td>True</td>
</tr>
<tr>
<td>int_filelength</td>
<td>File size of the downloaded document</td>
<td>0</td>
</tr>
<tr>
<td>txt_urldigest</td>
<td>Hash for the URL, for file storage purpose</td>
<td>&lt;null&gt;</td>
</tr>
<tr>
<td>bool_htmlparse</td>
<td>Indicate if the document has been parsed for new URLs</td>
<td>False</td>
</tr>
<tr>
<td>int_random</td>
<td>A random number to determine the download order. Avoid overloading the web server.</td>
<td>0</td>
</tr>
<tr>
<td>int_progress</td>
<td>Status of the document.</td>
<td>0</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>txt_url1</td>
<td>URL of the source document</td>
<td></td>
</tr>
<tr>
<td>txt_url2</td>
<td>URL of the target document</td>
<td></td>
</tr>
<tr>
<td>txt_file1</td>
<td>Filename of the source document</td>
<td></td>
</tr>
<tr>
<td>txt_file2</td>
<td>Filename of the target document</td>
<td></td>
</tr>
<tr>
<td>dbl_DTW</td>
<td>DTW score of the pair</td>
<td></td>
</tr>
<tr>
<td>dbl_r2</td>
<td>R2 score of the pair</td>
<td></td>
</tr>
<tr>
<td>bool_output</td>
<td>Status of output:</td>
<td>0</td>
</tr>
</tbody>
</table>

0 for unprocessed
1 for being downloaded
2 for downloaded
5 for ready to find parallel documents
6 for at least one parallel candidate found
7 for no parallel document found

**bool_finished** | True if the URL has been finished and will not be updated any more. | False |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>bool_tfidf</td>
<td>Indicate if the document has been trained for TF-IDF</td>
<td>False</td>
</tr>
<tr>
<td>int_refer</td>
<td>&lt;id&gt; of URL that linked to this document</td>
<td>0</td>
</tr>
<tr>
<td>time_nextaccess</td>
<td>Time for the next access when updating the web page.</td>
<td>&lt;null&gt;</td>
</tr>
<tr>
<td>int_txtlength</td>
<td>Size of extracted plain text of the document (HTML parsed file)</td>
<td>0</td>
</tr>
</tbody>
</table>

**Result Table**
1: processed but discarded
2: processed and outputted

Urlrules Table

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>txt_domain</td>
<td>Applied to domain name</td>
<td>&lt;null&gt;</td>
</tr>
<tr>
<td>txt_find</td>
<td>Regular express to be matched in URL</td>
<td>&lt;null&gt;</td>
</tr>
<tr>
<td>txt_replace</td>
<td>Matched string to be replaced by this string</td>
<td>&lt;null&gt;</td>
</tr>
<tr>
<td>bool_enabled</td>
<td>This rule is enabled if true</td>
<td>True</td>
</tr>
</tbody>
</table>

6. Database Operation

Add URL

Open urls table in MySQL database using MySQL Workbench installed with MySQL server. Type URL in txt_url and specify domain name of the URL in txt_domain.

Add New URL Rule

Open urlrules table in MySQL database using MySQL Workbench installed with MySQL server. Type domain name in txt_domain, find text in txt_find and replace text in txt_replace. Enable the rule by input 1 in bool_enable.
Select URLs for Parallel Document Mining

Please change `int_progress` from 2 (downloaded) to 5 (ready for parallel document mining) before executing PSM_FindParallel.jar.

Open `urls` table in MySQL database using MySQL Workbench installed with MySQL server. Execute the following SQL to make the change.

```
UPDATE `urls` SET int_progress=5 WHERE int_progress=2
```