Exploiting the Web as Parallel Corpora for Cross-Language Information Retrieval

Jian-Yun Nie, Jiang Chen

Département d'Informatique et Recherche Opérationnelle
Université de Montréal
C.P. 6128, succursale CENTRE-VILLE
Montréal (Québec), Canada H3C 3J7
Email: {nie, chen}@iro.umontreal.ca

The expansion of the Web creates more requirements for Cross-Language Information Retrieval (CLIR). Query translation is the key problem. Previous studies have shown that query translation can be done by exploiting a large set of parallel texts. However, the problem arisen is the unavailability of large parallel corpora for many languages. In this paper, we describe a mining system that automatically discovers parallel Web pages on the Web. This system exploits the existing search engines and the common characteristics in the organization of Web pages. Several large text corpora have been constructed using this system. This paper describes the mining process as well as the experimental results for English-French and English-Chinese CLIR. Our experiments show that query translation using the mined corpora can be as good as those by high-quality machine translation systems. This study shows the feasibility of building automatically a query translation system for all the active languages on the Web.

1. Introduction

Internet is becoming more and more multilingual in terms of both the documents published and the users: many documents on the Web are written in a language other than English, and many Internet users are non native English speakers. For example, a recent survey by China Internet Network Information Center shows that the number of Internet users in China increased 49.8% in 2001, and reached 33.7M at the end of 2001[1]. In addition, it is also shown that the international bandwidth is more than doubled in 2001 (reaching 7597.5 MB at the end of 2001). This shows that many Chinese users are interested in reading documents published outside of China, most of them written in English. Similar phenomena produce in several other countries.

For many users, the language barrier represents a serious problem. Although many users can read and understand a little bit English, they feel unease to formu-

[1] http://www.cnnic.net.cn/develst/rep200201-e.shtml

late queries in a foreign language (e.g. English). This is because of their limited vocabulary of English, or the possible misusage of English words. For example, a Chinese user may use "economic" instead of "cheap/economical/inexpensive" in a query because these words have similar translation in Chinese. An automatic query translation tool would be very helpful to these users.

On the other hand, even if a user knows well several languages, it is still a burden to formulate several queries in different languages. A query translation tool would also be very helpful.

The above description makes it clear that there is an increased requirement for Cross-Language Information Retrieval (CLIR), i.e. to retrieve relevant documents written in languages different from that of the query (without this latter being translated manually). Automatic query translation is the key problem of this task. This paper deals with one particular query translation method – the one that exploits parallel texts mined from the Web. Before describing our approach, let us first describe briefly the approaches that have been proposed in the literature.

1.1. Query translation

A Machine Translation (MT) system translates automatically a sentence/text from one language to another. Several such systems are available on the Web. For example, Systran[2] can translate texts between several pairs of European languages. At first glance, such a system seems to be the most appropriate tool for query translation. However, further analyses show that this solution may have several drawbacks.

On the one hand, the quality of the translation is often unsatisfactory. For the purposes of Information Retrieval (IR), it is not important that a query translation be a well-formed sentence because the syntactic aspects are not taken into account. Word selection and weighting are two important factors that have great impact on the retrieval effectiveness. In MT, the translation words selected are not always appropriate. In particular, MT systems often have problem to deal with ambiguous words such as "drug" in "drug traffic" and "drug administration office". For example, Systran suggests the same French translation "stupéfiant" (illegal product) for "drug" in both cases, while an English-Chinese MT system - ReadWorld[3] – suggests translating it as legal "medicine" in both cases. In addition, no weighting is created on the translation words. This means that all the translation words are virtually weighted equally.

On the other hand, MT usually selects only one translation word/term for each original word/term. In reality, there may be several reasonable translation words. For example, "computer" can be expressed in two common ways in Chinese: 计算机 and 电脑. It is better to translate "computer" with both terms in order to retrieve more relevant documents. This is known as the natural query expansion

---

[2] http://www.systransoft.com/

[3] http://www.readworld.com/

effect in query translation. However, the selection of a single word by an MT system prevents query translation from taking advantage of this natural benefit.

The most serious problem with MT is the unavailability of MT systems for many languages. It is costly to develop a new MT system for the required language pairs. In the near future, we cannot expect high-quality MT systems being built for many less common languages.

Two alternative approaches have been suggested in the literature to replace or complement the MT approach. One is based on the use of bilingual dictionaries, which are now widely available. However, several experiments have shown that a simple utilization of a bilingual dictionary cannot achieve a high effectiveness in retrieval [12]. The main problems are due to the poor coverage of the vocabulary and the difficulty to select appropriate translations (e.g. the "drug" case). There have been a number of studies that try to solve this problem by incorporating a statistical measure so as to select the translation word that is the most coherent with the context [6, 19]. In this paper, we will not focus on these methods.

The second alternative is based on an exploitation of large parallel text corpora, i.e. sets of texts with their translations. A parallel text corpus contains valuable translation knowledge. It has been exploited for the purposes of machine-aided translation. For example, translators often wonder about the translation of unusual or specialized expressions. An appropriate parallel corpus provides previous translations as examples from which the translator can inspire. TransSearch[4] is such a system based on parallel texts that provides the sentences containing possible translations of a given expression (query). One can go further in this direction by training a statistical translation model that provides translations for words/terms. The basic idea is to observe the co-occurrences of a source word (a word in the source language) and a target word (a word in the target language) in the parallel texts. The higher the frequency of co-occurrence, the more probable they are mutual translation (see section 3). As a result, a translation model can determine the probable translations of a given word, together with their probabilities. It can be directly used to query translation in CLIR: We can choose the most probable translation words as a "query translation".

It is obvious that the translation model cannot produce a grammatically correct translation. However, the grammatical aspects are not critical for the current search engines or information retrieval (IR) systems. What is important is a correct selection of translation words and an appropriate weighting in the translation. A statistical translation model is able to fulfill both tasks. Yet the construction cost for a statistical translation model is minimal because the training of the translation model can be fully automatic, provided that a parallel text corpus is available.

[4] http://www.TSrali.com

1.2. The need of parallel corpora

Most previous work on parallel texts has been conducted on a few manually constructed parallel corpora. The Hansard corpus is with no doubt the most used one. This corpus contains several years' debates in the Canadian parliament in both English and French, which amounts to several dozens of millions words in each language. The European parliament documents represents another large parallel corpus in several European languages. However, its availability is much more restricted than the Canadian Hansard. For Chinese and English, the Hong Kong government publishes official documents in both Chinese and English. They form a Chinese-English parallel corpus. However, its volume is much less than the Canadian Hansard. For many other languages, no large parallel corpora are available for the training of statistical models.

On the other hand, we observe that the increasing usage of different languages on the Web also result in more and more bilingual and multilingual sites. Many Web pages are translated into different languages. The Web contains a large number of parallel Web pages for many languages (usually with English). If they can be extracted, then the problem of parallel corpora can be solved.

In this study we attempt to mine the Web automatically for parallel Web pages. It is our goal to build an automatic mining system for parallel web pages for the purposes of CLIR.

In the remaining of the paper, we will first describe the principle and the organization of the mining process. Then the training of statistical models with the mined corpora will be described. The CLIR experiments with these models are described. Globally, the translation models trained on the mining results can produce query translations of comparable quality to those obtained with the best machine translation systems.

2. Mining for Parallel Texts - PTMiner

Although many parallel Web pages exist on the Web, it is not obvious to identify them and to confirm that a pair of pages is truly parallel. It would be a tedious work if we want to do it manually. Then how can this be done automatically? Of course, an automatic mining program is unable to understand the texts to judge if they are parallel. Nevertheless, several heuristic features provide useful indications. For example, if an English page points to another page with an anchor text "Chinese version", this is a useful indication that the second page is a Chinese version of the first page. These indications are not fully accurate, and they can produce errors. For the purpose of query translation, however, a noisy parallel corpus is still useful. This will be shown in our experiments in CLIR.

2.1. General principle of automatic mining

Parallel web pages often are not published in isolation. They are often connected in some way. For example, Resnik [15] observed that parallel Web pages often are referenced in the same parent index web page. In addition, the anchor text of such links usually identifies the language. For example, if a home page "`index.html`" contains links to both English and French versions of the next page, and that the anchor texts of the links are respectively "`English version`" and "`French version`", then the referenced pages are parallel. In addition, Resnik assumes that parallel Web pages have been indexed by large search engines existing on the Web. Therefore, in his approach, a query of the following form is sent to Alta Vista in order to first retrieve the common index page:

```
anchor: english  AND  anchor: French
```

Then the referenced pages in both languages are retrieved and considered to be parallel pages. Using this method, Resnik was able to mine a few small sets of parallel corpora: 2491 pairs of English-French Web pages, 3376 pairs of English-Chinese pages and 59 pairs of English-Basque pages[5].

We notice that only a small number of web sites are organized in this way. Many other parallel pages do not satisfy this condition. Our mining strategy uses different criteria. In addition, we also incorporate an exploration process (host crawler) in order to discover more web pages that have not been indexed by the existing search engines.

Our mining process is separated into two main steps: first identify as many candidate parallel pages as possible, then verify external features and contents to determine if they are parallel. Our mining system is called PTMiner (for Parallel Text Miner). The whole process is organized into the following steps (also see Fig. 1):

1. Determining candidate sites – This step tries to identify the Web sites where there may be parallel pages.
2. File name fetching – It identifies a set of Web pages from each Web page that are indexed by search engines.
3. Host crawling – It uses the URLs collected in the last step as seeds to further crawl each candidate site for more URLs.
4. Pair scanning by names – It makes the first pairing according to the similarity of the obtained URLs.
5. Content verification - The candidate parallel pages are further verified using the contents.

The steps 1-3 aim to find possible parallel pages, and the steps 4 and 5 try to verify if they are parallel. This mining process is based on the following two principles:

1. Exploiting the existing search engines as much as possible;
2. Using external features of web pages before comparing their contents.

Both aim to increase the efficiency of the process.

---

[5] http://umiacs.umd.edu/~resnik/strand/

- There are a number of large search engines on the Web, each indexing a large set of Web pages. They can be used to identify a first set of candidate web sites and web pages, from which further exploration is made. It is not a good idea to restart the work from stretch.
- External features are those that we can verify without downloading the file (e.g., URLs). Without comparing the contents of two pages, some external features can give good indication on whether they can be parallel. This fast verification is a preliminary step before a more costly content verification.

In the following subsections, we will present these steps in more detail.

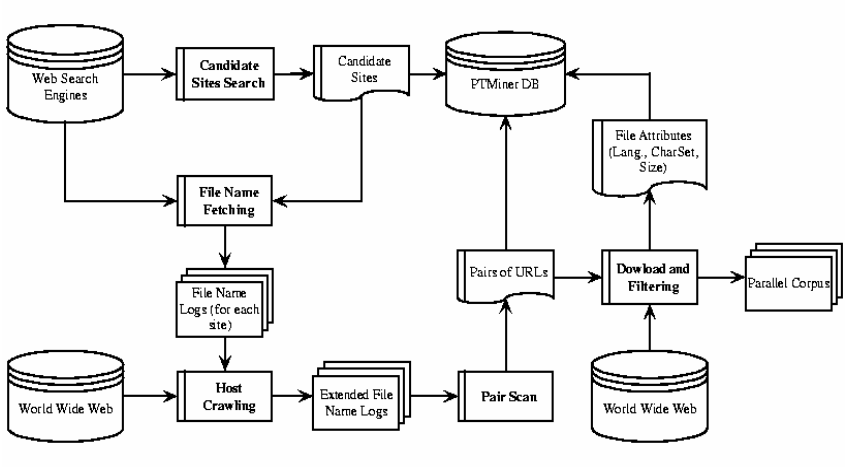

Fig. 1. The organization of PTMiner processes.

2.2. Identification of Candidate Web Sites

Parallel Web pages often cross-reference each other. For example, an English Web page often contains a pointer to the French version, and vise versa. In addition, the anchor text of these pointers often indicates clearly the language of the other page. For example, the anchor text of the pointer in the English page that points to the corresponding Chinese page may be “Chinese version”, “in Chinese”, and so on (Fig. 2 – first part). Another common organization is to set up a common index file that points to two different versions (Fig. 2 – second part). Still, the anchor text of the links also indicates the language. This second organization is considered by Resnik.

This clear language identification is helpful for readers to choose a version. It also indicates to PTMiner that the referred page may be the Chinese version of the page. Therefore, if there are such references links with language identification as anchor text, we consider that they are candidate of parallel pages. A Web site that contains at least one such candidate is a candidate Web site.

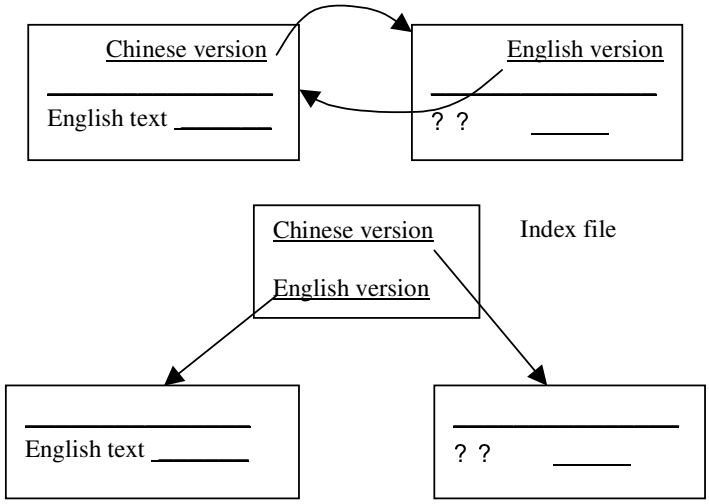

Fig. 2. Common organizations of parallel Web pages.

In order to determine the candidate sites, we take advantage of the large amount of Web sites indexed by the search engines. To the search engine AltaVista, we send a particular request asking for English pages that contain a link with an anchor text identifying another language[6]. For example:

```
anchor: chinese version, [in chinese, ...]
language: English
```

In the same way, we can obtain a second set of answers of Chinese pages containing pointers to an English version. From the union of two answer sets, we extract the URLs of Web sites, and they are considered as candidate sites.

Although we cannot cover all the possible candidate sites in this way, we can still determine a large set (several thousands) of web sites. If a larger number is needed, a robot has to be used [14].

[6] One can also send similar request to other search engines.

2.3. File Name Fetching

We assume that parallel pages are stored on the same Web site. This is not always true, but this assumption allows us to minimize the exploration of the Web, and to avoid considering many unlikely candidates.

To search for parallel pairs from each candidate site, PTMiner first asks the search engines for all the Web pages from this site they have indexed. This is done by a query of the following form:

```
host: <hostname>
```

If we only require a small number of parallel texts, this result may be sufficient. For our purpose, we need to explore the sites more thoroughly by a host crawler because:

- the search engines do not index all the Web pages of a site;
- most search engines allow users to retrieve a limited number of documents (e.g. 1000 in AltaVista).

Therefore, we continue our search with a host crawler, which uses the Web pages found by the search engines as seeds.

2.4. Host Crawling

A host crawler is slightly different from a Web crawler or a robot [14] in that a host crawler only exploits one Web site. A breadth-first crawling algorithm is used in this step. The principle is that if a retrieved Web page contains a link to an unexplored document on the same site, this document is added to a list that will be explored later. This crawling step allows us to obtain more web pages from the candidate sites.

2.5. Pair Scan by names

Once a large set of URLs is determined, the next task is to find out parallel pairs from them. One may directly compare the contents of the documents in order to determine if they are parallel. However, this would require the downloading of all the candidate documents. Many of them are indeed non-parallel. The useless downloading would create a heavy load on the network. In order to reduce the useless load of the network, we first verify the external features to eliminate the files that are unlikely to be parallel. The URL of a file is such an external feature.

We observe that many parallel pages have very similar file names. For example, an English web page with the file name “index.html” often corresponds to a French translation with the file name “index_f.html”, “index_fr.html”, and so on. The only difference between the two file names is a segment that identifies the language of the file. This similarity in file names is by no way an accident. In fact, if a Web site contains a large number of Web pages in different languages, the way to keep track of the documents in different versions is to give them the same

name, together with a language identification mark. This is the way in which people are encouraged to organize their web pages[7].

This same observation also applies to URL paths. In some cases, the two versions of the web page are stored in two different directories, for example:

"`www.asite.ca/en/afile.html`"
vs. "`www.asite.ca/fr/afile.html`".

So in general, a similarity in the URLs of two files is a good indication of their parallelism.

Therefore, we use the similarity of the URLs to make a preliminary selection of candidate pairs. Only the pairs with similar URLs are kept.

In terms of comparison algorithm, a straightforward method can be used to compare every couple of files. However, this method is inefficient and its complexity is quadratic. When we have to process thousands of files for each site, the computing time is long. Instead, we use the following pair-scanning algorithm: For each file name, we generate the possible corresponding file names for the other language, and check if such a file really exists. To do this, we define four lists of prefixes and suffixes for the two languages. For example:

English Prefix = {_, e, en, eng, engl, english, e,
en_, eng_, english_, ...}

Once a possible English prefix is identified in an URL, it is replaced by a Chinese prefix, and we test if this URL exists in our list. Although in this process, many variations of file name are checked, the computing time for each file is a constant. The whole processing time increases linearly with the number of the files.

2.6. Filtering by contents

The remaining file pairs are further verified by their contents. The following criteria may be used: file length, HTML structure, language verification, and sentence alignment.

2.6.1. Text Length

A pair of parallel pages usually has similar file lengths. A simple verification is then to compare the lengths of the two files. The only problem is to set a reasonable threshold that can filter out most wrong pairs without sacrificing too many good ones, i.e., balance between recall and precision. The typical length ratio depends on the language pair we are dealing with. For example, Chinese-English parallel texts usually are more different in length (about 1:2) than English-French ones (French texts are slightly longer, about 1:1.2). The filtering threshold has to be set from the actual observations. In our approach we tolerate a difference up to 40% with the typical ratio.

---

[7] See http://www.w3.org/Talks/1999/0830-tutorial-unicode-mjd/

2.6.2. HTML Structure and Alignment

Parallel web pages are usually designed to look in the same way. This often means that the two parallel pages have similar HTML structures. Therefore, the similarity in HTML tags is another filtering criterion.

However, we also notice that the HTML structures of parallel pages may be different. One of the reasons is that the two files may be created using two different HTML editors. This situation occurs more often when the two languages are very different (e.g. for English and Chinese). They look similar, but have different HTML markups. Therefore, certain flexibility is also allowed in this step. Another reason is that the two versions may be modified separately after their creation.

In our approach, we first determine a set of meaningful HTML tags, and extract them from both files (e.g. <p> and <H1>, but not <meta> and <font>). Meaningful tags are those that have an impact on the appearance of a Web page. A "diff"-style comparison will reveal how different the two sequences of tags are. A threshold is set to filter out the pairs that have a difference ratio higher than a threshold.

At this step, non-textual parts of the pages are also removed. If a page does not contain enough text, then the page is also removed.

2.6.3. Language and Character Set

When we query search engines for documents in one specific language (e.g. Chinese), the returned documents may be actually in a different language (e.g. Korean or Japanese). This is because the language of the documents has not been identified accurately. This identification is usually done automatically by using a language model (only some rare documents contain an identification of the language as a meta-data). As a consequence, the first set of documents we obtained from the search engines is not necessarily all in the required languages.

Our pair scan criterion also only exploits the name similarity of parallel pages. This is not a fully reliable criterion. Files with a segment "_en the files are in the required languages.

In our system, we use the SILC[8] system for an automatic language and encoding identification. SILC uses n-gram statistical language models to determine the most probable language and encoding schema for a text. It has been trained on several large corpora for each language. The accuracy of the system is very high. When a text contains at least 50 characters, its accuracy is almost perfect. By using SILC, a set of previously found file pairs are eliminated.

2.6.4. Filtering after Sentence Alignment

After the previous filtering steps, some non-parallel web pages still remain. We observe that many undesirable pairs cannot be correctly aligned at the sentence level, i.e. many sentences cannot be aligned with sentences in the other language. We call these alignments "empty alignments". If a sentence alignment results in

---

[8] See http://www-rali.iro.umontreal.ca/ProjetSILC.en.html

too many empty alignments (the proportion is higher than a threshold), the pair is considered to be non-parallel and removed. In our experiments, this threshold is set at 5% for Chinese-English corpus. This value seems to result in good translation quality.

Sentence alignment is also a necessary step before the training of statistical translation models. So we will present this process in more detail in a later section.

2.7. PTMiner implementation

PTMiner is implemented as a distributed system involving various processes in various machines. A centralized monitoring GUI interface is provided for user to watch clearly the working situation of all the processes, the content of the PTMiner database as well as the overall mining progress.

Fig. 3 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 packet).

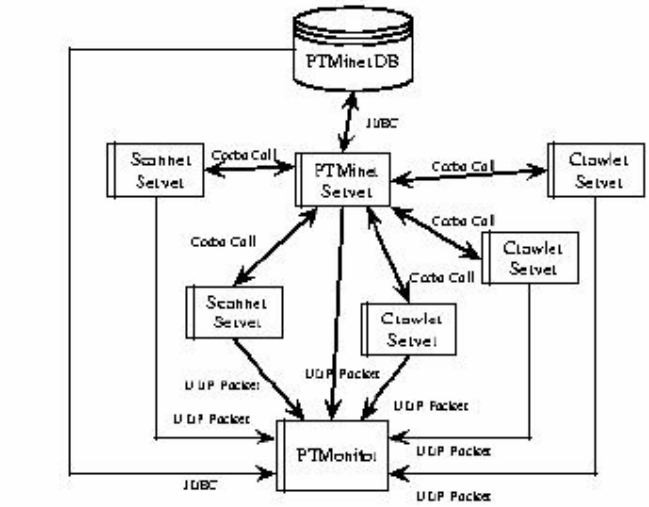

Fig. 3. PTMiner architecture

The central control unit of the system is the PTMiner server, which reads candidate sites from the database and assigns 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. Mining results are stored into the database. PTMonitor is a central GUI interface receiving messages

(in UDP packets) from all servers. It is also a viewer of the database content. Below is a brief description of the main modules and characteristics.

- PTMiner Database: The PTMiner database serves as the storage of intermediate and final mining results as well as working situation of the servers. For example, the "file" table contains the information of all the parallel pages, their URL, host, length, language and character set. The database is implemented with MySQL, a multi-threaded SQL database server.
- Site Fetcher: The site fetcher module is a stand-alone program (this is a step working off-line, not shown in Fig. 3) which implements the first step, candidate sites search, of the mining algorithm. As mentioned earlier, it sends queries to AltaVista and retrieves a set of candidate sites. The sites are stored into the database for future processing.
- Crawler Server: When a Crawler server is started, it first registers in the database and also notifies PTMonitor. It provides two methods that can be invoked by the PTMiner server, *fetch* and *crawl*. 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. The *crawl* method can be skipped if host crawling is not necessary.
- Scanner Server: Similarly to the Crawler server, the Scanner server registers itself in the database and sends messages to PTMonitor. Its *scan* method takes a site name, opens the corresponding file name log, and then scans for parallel pairs. The results are stored into the database.
- 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.
- PTMonitor: The objective of PTMonitor is to facilitate the monitoring of the whole mining process. It provides the user with various kinds of information including:
- Allocating System Resource: One advantage of the PTMiner system is that most of its modules are implemented in Java which enables them to run in practically any machine. This feature brings convenience in distributing working objects. Most modules of PTMiner consume very low (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. However, its actual working time on each site is much shorter than that of Crawler servers. In practice, we may need many Crawler servers but only one or two Scanner servers.

2.8. Generated Corpora

Despite a few language-dependent parameters such as prefix- and suffix lists, the mining process is relatively independent of particular languages, and can be used to any language. As a matter of fact, PTMiner has been successfully used to mine

parallel Web pages between English and French, Chinese, Italian and German with only minor adaptation. In this paper, we will focus on the two following corpora: English-French and English-Chinese.

The mined English-French corpus contains 14,198 pairs of parallel pages that are mined from about 30% of the 5,474 candidate sites identified. The mining process was stopped manually after 75 hours. The corpus actually includes 135MB French texts and 118MB English texts. Because there are many English-French parallel Web pages, host crawling was not used.

For English-Chinese, there are less bilingual sites. Therefore, host crawling is used in order to obtain a larger number of candidates. In our current experiment, we limited the mining domain in *hk* because Hong Kong is a perfect English-Chinese bilingual city where high quality parallel Web sites exist. 185 candidate sites have been searched. The resulted corpus contains 14,820 pairs of texts including 117.2MB Chinese texts and 136.5MB English texts. The mining process lasted about a week.

We examined a set of randomly selected pairs. It is estimated that over 95% of the pairs in the English-French corpus are truly parallel. In the case of English-Chinese, about 90% of the pairs are to be parallel.

3. Training Statistical Translation Models on Parallel Corpora

Most work on the training statistical translation models follow the models (called IBM models) proposed by Brown et al. [1]. In our case we use the IBM model 1. This model does not consider word order in sentences. Each sentence is considered as a bag of words. Any word in a corresponding target sentence is considered as a potential translation word of any source word. This consideration is oversimplified for the purpose of machine translation. However, for IR, as the goal of query translation is to identify the most probable words without considering the syntactic features, this simple translation model may suffice.

In order to train a translation model, parallel texts are usually decomposed into aligned sentences, i.e. for each sentence in a text, we know its translation sentence in the other language. The primary goal of producing sentence alignment is to reduce the scope of translation relationships between words: instead of considering a word in a source text to correspond potentially to every word in the target text, one can limit this relationship within the corresponding sentences. This allows us to take full advantage of the parallel texts and to produce a more accurate translation model.

3.1. Sentence alignment

Sentence alignment tries to create translation relationships between sentences. Sentences are not always aligned into 1:1 pairs. In some cases, one sentence can be translated into several sentences, and the sentence may even be deleted or a

new sentence may be added in the translation. This adds some difficulties in sentence alignment.

Gale & Church [5] is a classical algorithm based on length. It has been shown that this algorithm can successfully align the Canadian Hansard corpus, which is rather clean and easy to align. However, as pointed out by Simard et al. [16] and Chen [3], 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 [16].

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. Examples are generation/génération and financed/financé 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.

For the alignment of English-French corpus, this algorithm is used. In addition of the cognates defined by Simard *et al*., in our case, the texts also contain similar HTML tags. These tags are considered as additional "cognates" in the alignment algorithm. As a consequence, the best alignment is the one that also align HTML tags.

For Chinese-English texts, the concept of cognate does not apply. Wu [18] suggests to use a small dictionary to provide "lexical cues" in a similar way as cognates. However, the small dictionary was defined to align the Hong Kong legislature documents. Only specific correspondences such as "Mr." - 议员 (congressman) are included. For general sentence alignment, this dictionary would have little impact. We extended this approach by including a large bilingual dictionary. Our experiments show that when the size of dictionary increases, the alignment quality also increases. The bilingual dictionary is incorporated in our alignment method to provide "known translation words" – the words in the target sentence that we can recognize as correct translations of source words. This method is inspired from human alignment process: If a human being knows well one language (e.g. source language), but only a few words in the target sentence, he/she would still be able to judge whether the target sentence can be translation of the source sentence by trying to align the known target words with the source words. In our approach we exploit the same idea: If the target sentence contains a percentage P of known translation words, then the alignment score is increased by P*T, where T is a parameter that denotes the importance of this factor. Our later experiments show that the best value of T is 1.5.

As we mentioned earlier, for the Chinese-English corpus, a further filtering is made according to the result of sentence alignment. If the proportion of empty alignments is higher than a certain threshold, then the text pair is removed. In our experiments, as the threshold increases, the resulting corpus becomes better (i.e.

both the translation accuracy and CLIR effectiveness are better- see sections 5 and 6).

3.2. Processing of words

The results of text alignment are two sets of sentences and the mapping data between them. We have now to segment Chinese sentences into words because Chinese sentences are written as continuous strings. This involves Chinese word segmentation as well as the transformation of words into a standard form for English and French. This last process aims to reduce mismatch during retrieval due to slight variations in word forms.

In the past decade, many Chinese segmentation approaches were studied. Two main categories are the dictionary-based approaches (e.g. [10]) and the statistical approaches (e.g. [17]). 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. [11] proposed an approach, which flexibly incorporates statistical information (if available) with dictionaries and heuristic rules.

Globally, the segmentation accuracy of both methods is comparable. They usually achieve an accuracy higher than 90%. For IR, a slight difference in word segmentation accuracy does not have a great impact on IR effectiveness. Therefore, we use the dictionary-based approach in this study. The dictionary we use contains 187,182 words/terms. Many entries are in fact phrases or compound terms. The including of these long words or phrases is useful to IR. It allows us to achieve at a higher precision.

However, it has been shown that we cannot apply the longest-matching strategy as one usually does in Chinese word segmentation. This is because a long word can contain short words. If we only extract long words, but not the implied short words, the recall will suffer. So we extract not only the long words, but also the short words included in the long words. For example, if “ABCD”, “AB” and “CD” are words, they will all be extracted from the string “ABCD”. This approach is proven to work well for Chinese IR [13].

For French and English, all the words and terms are transformed into a standard citation form. For example, all the verbs are transformed into its infinitive form, and all the nouns into singular form. This transformation is based on a statistical tagging of English and French.

In addition stopwords are also removed from the corpora. If they were not removed, the resulting translation models would often suggest stopwords as prob-

able translations for meaningful source words because stopwords have very high frequency of occurrences in the parallel corpora.

3.3. Model training

The principle of model training is: in a set of aligned sentences, if a target word *f* often co-occur with a source word *e* in the aligned sentences, then there is a high chance that *f* is a translation of *e*, i.e. the translation probability *t*(*f*|*e*) is high. The training algorithm uses dynamic programming to determine a probability function *t*(*f*|*e*) such that it maximizes the expectation of the given sentence alignments (see [1] for details).

The training of statistical models follows the models proposed by Brown et al. [1]. The principle 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 aligned sentences) the word translation probability *t*(*f*|*e*) that a target word *f* is a translation word for a source word *e*.

We briefly describe the training for IBM model 1 as follows.

The translation probability function *t* is determined such as to maximize the probability of the given sentence alignments *A* of the training corpus. Suppose a sentence alignment $\mathbf{e} \leftrightarrow \mathbf{f}$, and that

$\mathbf{e}=\{e_1, e_2, e_3, ..., e_l\}$,

$\mathbf{f}=\{f_1, f_2, f_3, ..., f_m\}$

where l and m are respectively the length of these sentences. Then the function *t* is:

$$t = \arg\max_t p(A)$$

$$= \arg\max_t \prod_{\mathbf{e}\leftrightarrow\mathbf{f}} p(\mathbf{f} \mid \mathbf{e})$$

$$= \arg\max_t \prod_{\mathbf{e}\leftrightarrow\mathbf{f}} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j \mid e_i)$$

The constraint is:

$$\sum_{f_j} t(f_j \mid e_i) = 1.$$

The determination of *t* can be done by applying iteratively EM (Expectation maximization) algorithm. We do not give details here. Interested readers can refer to [1].

As mentioned earlier, IBM model 1 considers every word in the target sentence to be equivalently possible translation of any word in the source sentence, regardless to their position and to the "fertility" of each word (e.g. an English word may be translated by one or more French words). It is obvious 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 is able to determine the word translation probability *t* between words, and this fits the

need of cross-language information retrieval of finding out the most important translation words.

4. Evaluation of the translation models

This evaluation has been done only for Chinese-English translation models (in both directions). 200 English and Chinese words are randomly selected and used for the evaluation. We examine manually the first (the most probable) translation proposed by the translation model, and determine if it is a good translation. The Chinese-English model was found to have a precision of 77%. The English-Chinese model has a higher precision of 81.5%. The following table shows some samples of the evaluation.

Table 1. Samples of word translation by the models.

English word	Chinese Translation	Probability	Correctness
a.m.	上午	0.201472	Yes
access	公开	0.071705	No (open)
adaptation	适应	0.179633	Yes
add	补充	0.317435	Yes
adopt	采用	0.231637	Yes
agent	代理人	0.224902	Yes
agree	同意	0.365690	Yes
airline	航空公司	0.344001	Yes
amendment	修订	0.367518	Yes

Table 2. Samples of word translation by the models.

Chinese word	English Translation	Probability	Correctness
办事处	office	0.375868	Yes
保护	protection	0.343071	Yes
报告	report	0.358592	Yes
备	prepare	0.189513	Yes
本地	local	0.421837	Yes
便会	follow	0.023685	No (will)
标准	standard	0.445453	Yes
补校	adult	0.044959	No (adult school)
不足	inadequate	0.093012	Yes

Globally, the precision achieved is relatively high. If such a translation model is used for MT it is obviously not good enough. However, for CLIR such a precision may be acceptable.

At this stage, we can also examine the impact of corpus filtering after sentence alignment on the quality of the translation models. The following table shows the accuracy of the models trained on a corpus with and without filtering. We can see that filtering after sentence alignment produces a better translation accuracy.

Table 3. Translation accuracy with and without filtering

Direction	No filtering	With filtering
E-C	80.50%	91.50%
C-E	77.00%	86.50%

5. CLIR Experiments

We use a translation model as follows in our query translation process: Each query word is submitted to the translation model, and this later will suggest the translation words together with their probabilities. Then the sets of translation words for all the query words are grouped, and the probabilities of the same translation word summed up. Finally, the N most probable translation words are kept as the query translation. In our earlier experiments [12], several values of N have been tested, and it turned out that a value between 25 and 50 produces the best results. The results reported in this paper are obtained with N=50.

Notice that all the queries in the test collection are provided in both the source and target languages. This allows us to compare CLIR effectiveness with monolingual IR effectiveness. The effectiveness of CLIR with a set of translated queries is measured in terms of average precision.

The basic monolingual retrieval system is using the Smart [2]. A minor change is made so that the system can accept a list of weighted words as input query.

5.1. English-French CLIR

The experiments described here are all conducted on the test corpora of TREC [7]. The experiments for English-French CLIR are conducted on English AP collection (242,918 documents) and French SDA collection (141,656 documents) used in TREC6 and TREC7. Both collections are newspaper articles. There are respectively 25 and 28 queries in TREC6 and TREC7.

The model trained on the mined web pages (not filtered by sentence alignment for English-French corpus) is called “Web model”. In addition, we also trained another model with a manually constructed parallel corpus – the Hansard (called Hansard model). For comparison, a MT system – Systran, and a bilingual dictionary are also used for query translation. In the case of dictionary translation, each query word is translated by all the translations stored in the dictionary. The effectiveness in each case is shown in the following tables (where French-English means translating French queries into English and then retrieving English documents).

Table 4. French-English CLIR results.

	F-E (%mono) Trec-6	F-E (%mono) Trec-7
Monolingual IR	0.2865	0.3202
MT translation	0.3098 (107.0%)	0.3293 (102.8%)
Dictionary	0.1707 (59.0%)	0.1701 (53.1%)
Hansard model	0.2166 (74.8%)	0.3124 (97.6%)
Web model	0.2389 (82.5%)	0.3146 (98.3%)

Table 5. English-French CLIR results.

	E-F (%mono) Trec-6	E-F (%mono) Trec-7
Monolingual IR	0.3686	0.2764
MT translation	0.2727 (74.0)	0.2327 (84.2%)
Dictionary	0.2305 (62.5%)	0.1352 (48.9%)
Hansard model	0.2501 (67.9%)	0.2587 (93.6%)
Web model	0.2504 (67.9%)	0.2289 (82.8%)

In both tables, we can observe that the Web models globally produce a similar performance to the Hansard models. This fact shows that an automatically mined parallel corpus is as effective as a manually constructed parallel corpus for the purpose of CLIR.

We can also observe that the effectiveness of both the Web models and the Hansard models is close to that of the MT system.

This result is extremely encouraging. It shows that potentially, we will be able to construct automatically inexpensive query translation tools for all the language pairs for which there are enough parallel Web pages, and these tools can be almost as good as the best MT systems.

5.2. English-Chinese CLIR

For English-Chinese CLIR, English document collection is the same AP corpus with 53 queries. The Chinese document collection is that used in TREC5 and TREC6 Chinese track (164,811 documents and 54 queries). It contains newspaper articles from the People's Daily and Xinhua News Agency. In this case, we also combined the translations by the Web model and those with the bilingual dictionary [4]. The MT English-Chinese translation is made with an online translation system Readworld[9].

The experimental results are shown in the following tables.

[9] http://www.readworld.com/tran/index.html

Table 6. English-Chinese CLIR results.

	C-E (%monolingual)	E-C (%monolingual)
Monolingual IR	0.3861	0.3976
MT translation	(Not available)[10]	0.2001 (50.3%)
Dictionary	0.1530 (39.6%)	0.1427 (35.9%)
Web model	0.2063 (53.43%)	0.2013 (50.63%)
Web model + Dictionary	0.2811 (72.81%)	0.2601 (65.42%)

These results further confirm that the Web models perform as good as an MT system. In particular, when the Web models are combined with a bilingual dictionary (with a fixed weight to a dictionary translation), the performance is even better.

The Web model shown in the above table is the one trained with the filtered corpus after sentence alignment. If the filtering is not used, then the effectiveness is lower: respectively 0.1654 and 0.1591 for C-E and E-C when the Web models are used alone, and 0.2583 and 0.2232 when they are used in combination with the dictionary. Again, we can see the great positive impact of the further filtering of the corpus on CLIR effectiveness.

6.3. Discussions

During our experiments, we observed several problems in query translation.

Non-translated sentences in mined training corpus

We observe that despite all the filtering criteria, the remaining web pages are not always parallel. This fact is a source of much translation noise. In a number of cases, Web pages are not completely translated. As a consequence, the resulting translation model may suggest words of the same language as a translation. In most cases, these words are function words (e.g. prepositions) that can be easily filtered out. This problem will be further investigated in the future by applying the SILC system to detect the language of each sentence.

Compound terms

The translation models generally have difficulties to translate compound words. For example, for the French term “pomme de terre” (potato), besides the correct translation, it is translated as “apple”, “earth”, “soil”, and so on (because the French term means “apple of soil”). The problem is due to the fact that the models make a word-by-word translation. A solution to this problem is to group the com-

[10] We do not have a Chinese to English MT system to compare with. However, in [9], a comparison showed that the CLIR effectiveness with the translation of an MT system (TransPerfect) is 56% of that of the monolingual IR. If the MT translation is also combined with an additional dictionary, the effectiveness is 62% of monolingual IR.

pound words into a single unit before model training, so that the translation model will be able to consider “pomme de terre” as a whole.

Chinese political terms and abbreviations

The sharp contrast of the CLIR performances between English-French and English-Chinese is not surprising. Chinese and English are two languages much more different than it is the case between English and French. In particular, it is difficult to translate important proper nouns between Chinese and English. For example, 皮纳图博火山 (Mount Minatubo), 苏比克 (Subic Bay), 南沙 (the Spratly Islands), 大亚湾 (Daya Bay) are all considered as unknown words. Political terms and abbreviations such as 三乱 (three turmoils), 一国两制 (one-nation-two-systems), 京九铁路 (Beijing-Kowloon railways) are also difficult to translate. None of the approaches using MT systems, dictionaries or statistical translation models can translate them correctly. This problem is much less serious between European languages because most proper nouns are written in the same way in different languages. The large difference between the two sets of experiments is largely due to the occurrences of proper nouns and abbreviations in the queries of the English-Chinese CLIR experiments.

Vocabulary coverage

In query translation, we did not observe that the coverage of vocabulary was a problem. This is because the words used in the test queries are quite common words, except for a few unusual words. For example, an unusual French word “maltraitance” is used in one query to express “(child) abuse”. None of the translation approach is able to recognize this word.

Difficulty to disambiguate words

Although the translation models can usually suggest the most used sense for an ambiguous word, it is unable to eliminate the other senses in the translation. For example, the word “drug” will be translated in both senses, with a higher probability for the sense “medicine” because more documents relate to this meaning. Our current utilization does not provide a context–sensitive query translation. The word “drug” will still be translated in the same way in “drug traffic” and “drug administration office”. For a context-sensitive translation, one may incorporate a language model in the translation. The goal if to choose the translation word that is the most coherent with the translation words of other source words. This approach has been successfully used in [6]. It can also be incorporated in our approach. This aspect will be investigated in our future experiments.

6. Conclusions

The increasing usage of different languages on the Web not only creates increased requirement for Cross-Language Information Retrieval tools, but also provides a

new possibility to automatically construct parallel corpora. This study described an automatic mining approach for parallel web pages. The mining process is based on heuristic rules derived from the common organization approaches of Web pages. It takes advantage of the existing search engines to identify candidate Web sites and Web pages, and gradually filter out non-parallel pairs by first applying criteria on external features, then on contents.

The system has been successfully applied for several language pairs: English-Chinese, English-French, English-German, English-Italian, and so on. Several experiments [8] have confirmed that these corpora are highly useful for query translation, and we can obtain an effectiveness comparable to that of an MT system. Yet their construction is minimal. The approach can be easily extended to other languages that are active on the Web. This opens the perspective of automatically constructing a query translation system on the Web for all the active languages on the Web.

References

1. Brown, P.F. Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L. (1993): The mathematics of machine translation: Parameter estimation. Computational Linguistics, 19:263-311.
2. Buckley, C. (1985): Implementation of the SMART information retrieval system. Cornell University, Tech. report 85-686.
3. Chen, S. F. (1993): Aligning sentences in bilingual corpora using lexical information. Proc. ACL, pp. 9-16.
4. Denisowski, Paul (1999): Cedict (chinese-english dictionary) project. (http://www.mindspring.com/ ~paul_denisowski/cedict.html).
5. Gale, W.A., Church, K.W. (1991): 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.
6. Gao, J., Nie, J.-Y., Xun, E., Zhang, J., Zhou, M., Huang, C. (2001): Improving Query Translation for CLIR using Statistical Models, 24th ACM-SIGIR, New Orleans, pp. 96-104.
7. Harman D. K., Voorhees, E. M. (eds.) (1997): *The Sixth Text REtrieval Conference (TREC-6).* Gaithersburg, NIST Special Publication 500-240 (http://trec.nist.gov).
8. Kraaij, W. (2001): TNO at CLEF-2001: Comparing translation resources, Workshop of Cross-Language Evaluation Forum (CLEF), Darmstadt, pp. 29-40.
9. Kwok, K. L. (1999): English-Chinese cross-language retrieval based on a translation package, *Workshop of Machine Translation for Cross Language Information Retrieval*, *Machine Translation Summit VII*, Singapore, pp. 8-13.
10. Liang, N. Y., Zhen, Y. B. (1991): A Chinese word segmentation model and a Chinese word segmentation system PC-CWSS. In COLIPS'91, volume 1, pages 51-55.

11. Nie, J.-Y., Jin, W., Hannan, M.-L. (1994): A hybrid approach to unknown word detection and segmentation of Chinese. In International Conference on Chinese Computing, pages 326-335, Singapore.
12. Nie, J.-Y., Simard, M., Isabelle, P., Durand, R. (1999): Cross-language information retrieval based on parallel texts and automatic mining parallel texts from the Web. In ACM SIGIR'99, pages 74-81.
13. Nie, J.-Y., Gao, J., Zhang, J., Zhou, M. (2000): On the use of words and n-grams for Chinese information retrieval. In: Fifth International Workshop on Information Retrieval with Asian Languages, IRAL-2000. Hong Kong.
14. Prosise J. (1996): Crawling the Web, A guide to robots, spiders, and other shadowy denizens of the Web, PC Magazine - July (http://www.zdnet.com/pcmag/issues/1513/pcmg0045.htm).
15. Resnik, Philip (1998) Parallel stands: A preliminary investigation into mining the Web for bilingual text, *AMTA'98*, *Lecture Notes in Artificial Intelligence, 1529,* October.
16. Simard, M., Foster, G.F., Isabelle, P. (1992): Using cognates to align sentences in bilingual corpora, *Proceedings of TMI-92*, Montreal, Quebec, pp. 67-81
17. Sproat, R., Shih, C. (1991): A statistical method for finding word boundaries in Chinese text. Computer Processing of Chinese and Oriental Languages, 4(4):336-351, 1991.
18. Wu, D. (1995): Large-scale automatic extraction of an English-Chinese lexicon, *Machine Translation*, 9(3-4): 285-313
19. Xu, J. Weischedel, R., Nguyen, C. (2001): Evaluating a probabilistic model for cross-lingual information retrieval, ACM-SIGIR, pp. 105 - 110