Cross-Language Information Retrieval

Jian-Yun Nie[1]

***Abstract*—A research group in University of Montreal has worked on the problem of cross-language information retrieval (CLIR) for several years. A method that exploits parallel texts for query translation is proposed. This method is shown to allow for retrieval effectiveness comparable to the state-of-the-art effectiveness. A major problem of this approach is the unavailability of large parallel corpora. To solve this problem, a mining system is constructed to automatically gather parallel Web pages. The mining results are used to train statistical translation models.**

When a query is translated word by word, the accuracy may be low. In order to increase the translation accuracy, compound terms are extracted and incorporated into the translation models, so that compounds can be translated as a unit, rather than as separate words. Our experiments show that this can further increase the CLIR effectiveness.

I. Introduction

INFORMATION retrieval (IR) tries to identify relevant documents for an information need, expressed as a query. The problems that an IR system should deal with include document indexing (which tries to extract important indexes from a document and weigh them), query analysis (similar to document indexing), and query evaluation (i.e. matching the query with the documents). Each of these problems has been the subject of many studies in IR.

Traditional IR identifies relevant documents in the same language as the query. This problem is referred to as monolingual IR. Cross-language information retrieval (CLIR) tries to identify relevant documents in a language different from that of the query. This problem is more and more acute for IR on the Web due to the fact that the Web is a truly multilingual environment. In addition to the problems of monolingual IR, CLIR is faced with the problem of language differences between queries and documents. The key problem is query translation (or document translation). This translation raises two particular problems [6]: the selection of the appropriate translation terms/words, and the proper weighting of them. In the last few years, researchers have worked on these problems intensively. Three main techniques for query translation have been proposed and tested:

- With an on-the-shelf machine translation (MT) system;
- With a bilingual dictionary;
- Or with a set of parallel texts.

[1] Département d'Informatiqueet Recherche opérationnelle, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Québec, H3C 3J7 Canada, nie@iro.umontreal.ca

The first two approaches are quite straightforward. We will not give details about them. Our research efforts have been concentrated on the third approach. This approach is promising because it does not require extensive manual preparation (in comparison with the construction of an MT system); and its translation is usually more appropriate than with a bilingual dictionary.

The major advantages of this approach are the following ones:

The training of a translation model can be completely automatic. No (or little) manual preparation is required.

The resulting translation model reflects well the word usage in the training corpus. This offers the possibility to train specialized and up-to-date translation models.

In this paper, we will describe our approach to CLIR based on parallel texts, as well as some experiments. The paper will be organized as follows. In Section II, we will first describe briefly the training process of statistical translation models on a set of parallel texts. Then we will describe in Section III the IR system we use for our experiments. Section IV describes our experiments with the translation models trained on a manually prepared parallel corpus. Section V describes our approach to mining parallel Web pages, as well as their utilization for CLIR. Section VI presents our utilization of compound terms in CLIR. Finally, we present our conclusions in Section VII.

II. Training Statistical Translation Models on Parallel Texts

Let us first describe briefly the training of statistical translation models on a set of parallel texts. These models will be used in our experiments.

Statistical translation models are trained on parallel texts. A pair of parallel texts is two texts which are translation one of the other. Model training tries to extract the translation relationships between elements of the two languages (usually words) by observing their occurrences in parallel texts. Most work on the training statistical translation models follows 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 determine its translation sentence(s) 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.

A. 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] propose an algorithm based on sentence length. It has been shown that this algorithm can successfully align the Canadian Hansard corpus (the debates in the Canadian House of Commons in both English and French), which is rather clean and easy to align. However, as pointed out by Simard *et al.* [12] 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 [12].

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.

Another method incorporates a dictionary [3]. The translations contained in the dictionary serve as cues to sentence alignment: a sentence is likely to align with another sentence if the latter contains several dictionary translations of the words of the former.

In our implementation, we use the approach of Simard *et al.* [12].

B. 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).

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 the sentences $\mathbf{e}$ and $\mathbf{f}$ are composed of set of words as follows:

$$\mathbf{e}=\{e_1, e_2, e_3, ..., e_l\},$$
$$\mathbf{f}=\{f_1, f_2, f_3, ..., f_m\}$$

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

$$\begin{aligned} t &= \arg\max_t p(A) \\ &= \arg\max_t \prod_{\mathbf{e}\leftrightarrow\mathbf{f}} p(\mathbf{f}\mid\mathbf{e}) \\ &= \arg\max_t \prod_{\mathbf{e}\leftrightarrow\mathbf{f}} \varepsilon(1+l)^{-m} \prod_{j=1}^{m}\sum_{i=0}^{l} t(f_j \mid e_i) \end{aligned}$$

where ε is the probability that an English sentence of length l can be translated into a French sentence of length m, and $t(f_j|e_i)$ the word translation probability of e_i by f_j.

The probability t can be determined by applying the iterative EM (Expectation maximization) algorithm. We do not give details here. Interested readers can refer to [1].

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.

III. IR SYSTEM

In our experiments we use the SMART system. SMART is an IR system, developed at Cornell University [2]. The indexing process considers every token as an index. Indexes are weighted according to the *tf*idf* weighting scheme[2]. This is a common way to weigh the importance and specificity of a term in a document. The principle is as follows: 1) the more a word occurs in a document, the more it is important. This is the *tf* factor. On the other hand, the more there are documents containing the word, the less the word is specific to one particular document. In other words, the word does not allow distinguishing a document from the others. Therefore, the weight of the word is lowered. This is the *idf* factor. More precisely, the two factors are measured as follows:

[2] tf = term frequency, and idf = inversed document frequency.

$$tf(t, D) = \log(freq(t, D) + 1);$$
$$idf(t) = \log(\frac{N}{n(t)})$$

where *freq*(*t, D*) is the frequency of occurrences of the word/term *t* in the document *D*; *N* is the total number of documents in the collection; *n*(*t*) is the number of documents containing *t*.

The retrieval process follows the vector space model [2]. In this model, a vector space is defined by all the tokens (words or terms) encountered in the documents. Each word/term represents a distinct dimension in this space. Then a document, as well as a query, is represented as a vector in this space. The weight in a dimension represents the importance of the corresponding word/term in the document or query (the *tf***idf* weight). The degree of correspondence between a document and a query is estimated by the similarity of their vectors. One of the commonly used similarity measures is as follows:

$$sim(D, Q) = \frac{\sum_i d_i \times q_i}{\sqrt{\sum_i d_i^2 \times \sum_i q_i^2}}$$

where d_i and q_i are respectively the weights of a term in the document *D* and in the query *Q*.

IV. Experiments with the Hansard models

There are a few manually constructed parallel corpora. The best known is the Canadian Hansard, which contains the debates of the Canadian parliaments during 7 years, in both French and English. It contains dozens of millions words in each language. Such a parallel corpus is a valuable resource that contains word/term translations. Our first experiments are carried out with translation models trained on the Hansard corpus- we call the resulting models the Hansard models.

We used two test collections developed in TREC[3], one in English (AP) and the other in French (SDA). Both collections contain newspaper articles. The SDA contains 141,656 documents, and AP 242,918 documents. We use two sets of about 30 queries, available in both French and English. These queries have been used in TREC6 and TREC7 for French-English CLIR. The queries have been manually evaluated (i.e. we know their relevant documents). Table I shows the CLIR effectiveness obtained with these translation models. F-E means using French queries to retrieve English documents, i.e. the French queries are first translated into English, then the English translation is used to match the documents. In all our experiments, we select the 25 most probable translation words as the "translation" of a query.

In Table I, the effectiveness is measured by average precision, i.e. the average of the precisions over 11 points of recall. This is a standard measure used in IR. We also show the percentage of the CLIR effectiveness with respect to the monolingual IR effectiveness (%mono). In comparison with the state-of-the-art effectiveness, which is usually around 80-90% of the monolingual effectiveness (see the reports of TREC at http://trec.nist.gov), the results we obtained are quite comparable.

TABLE I.
AVERAGE PRECISION USING HANSARD MODEL

	F-E (%mono)	E-F (%mono)
Trec6	0.2166 (74.8%)	0.2501 (67.9%)
Trec7	0.3124 (97.6%)	0.2587 (93.6%)

V. Mining of parallel Web pages

A major problem to use parallel texts is often the unavailability of large parallel corpora. In order to obtain such corpora, we constructed a mining system – PTMiner [4] – to automatically gather parallel Web pages.

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. In our mining approach, we exploit several heuristic features. For example, if an English page points to another page with an anchor text "French version" or "version française", this is a useful indication that the second page is a French version of the first page. Although these indications are not fully accurate, and they can produce errors, we will show later in our experiments that a noisy parallel corpus is still useful for query translation in CLIR.

In the following subsections, we will briefly describe our mining approach.

A. Automatic Mining

Parallel web pages often are not published in isolation. Most of the time, they are connected in some way. For example, Resnik [11] 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.

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

[3] TREC: Text Retrieval Conference, a series of conferences aiming to test IR systems with large document collections. See http://trec.nist.gov/

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:

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 pairs the Web pages according to the similarity of their URLs.

IDENTIFICATION OF CANDIDATE WEB SITES

To determine candidate sites, we assume that a candidate site contains at least one page that refers to another version of the page, and the anchor text of the reference clearly identifies the language. For example, an English Web page contains a link to the French version, and the anchor text is "French version", "in French", "en français" and so on. So to determine the candidate sites, we send a particular request to search engines asking for English pages that contain a link with an anchor text identifying another language such as:

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

The host addresses we extract from the resulting Web pages correspond to the candidate sites.

FILE NAME FETCHING

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>
```

However, a search engine may not index all the Web pages on a site. To obtain a more complete list of URLs from a site, we need to explore the sites more thoroughly by a host crawler.

HOST CRAWLING

A host crawler is slightly different from a Web crawler or a robot [10] 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.

PAIR SCANNING BY NAMES

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 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. This similarity is used to make a preliminary selection of candidate pairs.

FILTERING AFTER DOWNLOADING

The remaining file pairs are downloaded for further content verification according to the following criteria:.

- Length of the pages: A pair of parallel pages usually has similar file lengths. A simple verification is then to compare the lengths of the two files. Note that the length ratio changes between different language pairs.
- HTML structure: Parallel web pages are usually designed to look similarly. This often means that the two parallel pages have similar HTML structures. Therefore, the similarity in HTML tags is another filtering criterion.
- The pair-scanning criterion we used only exploits the name similarity of parallel pages. This is not a fully reliable criterion. Files with a segment "en_" may be not in English. Therefore, a further verification is needed to confirm that the files are in the required languages. In our system, we use the SILC4 system for an automatic language and encoding identification.

With PTMiner, we have been able to collect several parallel corpora from the Web. Table II shows some of them.

TABLE II.
SIZES OF THE WEB CORPORA

	FR-EN		DE-EN		IT-EN	
# Text Pairs	18 807		10 200		8 504	
Raw data (MB)	198	174	100	68	50	77
Cleaned data (MB)	155	145	66	50	35	50

In our further description, we will concentrate on the French-English pair.

B. CLIR With the Web Models

Translation models are trained on the set of parallel Web pages as described in Section 2, except that some preprocessing has to be performed on these pages in order to remove HTML tags. Once translation models (in both directions) are trained, they are used to produce 25 most probable translation words that are considered as the translation of a query. Table III describes the CLIR

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

effectiveness with the Web models.

TABLE III.
AVERAGE PRECISION USING WEB MODEL

	F-E (%mono)	E-F (%mono)
Trec6	0.2103 (72.6%)	0.2595 (70.4%)
Trec7	0.2380 (74.3%)	0.1975 (71.5%)

In comparison with the Hansard model, we see that the Web models perform slightly worse. However, considering the noise that this training corpus may contain, this effectiveness is quite good. It is still close to the state-of-the-art effectiveness. This test shows that the automatically mined parallel Web pages are greatly useful for CLIR.

VI. INCORPORATING COMPOUND TERMS IN TRANSLATION MODELS

In the previous approach, parallel texts have been exploited to find translations between single words. The most obvious problem we can see is that by taking words one by one, many of them become ambiguous. The translation model will then suggest several translations corresponding to different meanings of the word. For example, the word "information" (in French) will have many possible translations because 1) the word denotes several meanings; 2) it appears very frequently in the parallel corpus. Among the possible translations, there are "information", "intelligence", "espionage", etc. However, if the term we intend to translate is "système d'information" (information system), and if the term is translated as a whole, then many of the meanings of "information" can be eliminated. The most probable translation of this term will be the correct term "information system". Through this example, we can see that a translation model that integrates the translation of compound terms can be much more precise. This is the goal of our utilization of compounds during query translation.

To do this, we have to train a translation model that incorporates compound terms as additional translation units to words. So compound terms are first extracted from the training parallel corpus, and added to the original sentences. Then the same translation process is launched. The resulting model contains the translations for both single words and compound terms.

To identify compound terms, we use both a large terminology database containing almost 1 million words and terms, and an automatic extractor of compound terms. The extractor uses syntactic structures, together with a statistical analysis. First, word sequences corresponding to predefined syntactic templates are extracted as candidates. If the frequency of occurrences of a candidate is above a certain threshold, then the sequence is considered as a compound term.

The first problem is the definition of the syntactic templates. This is done manually according to the general knowledge on syntactic structures of a language. Usually the extraction is restricted to noun phrases. For example, the following template is used in the tool we used - Exterm:

```
((NC|AJ) )*((NC|AJ)|NC PP) ((NC|AJ) )*NC
```

where NC means a common noun, AJ an adjective, and PP a preposition.

Of course, a POS (Part-Of-Speech) tagging is necessary in order to recognize the syntactic category of each word. The tagger we used is a statistical tagger trained on the Penn Treebank[5]. It tries to determine the most probable syntactic categories that fit the best the words of a sentence. Details on the training of such a tagger can be found in [7].

All the terms and words in documents, queries and the training parallel corpus are submitted to a standardization process on words, as follows:

- Nouns in plural are transformed into singular form (e.g. systems → system);
- Verbs are changed into infinitive form (e.g. retrieves → retrieve, retrieving → retrieve);
- Articles in a term is removed (e.g. the database system)

For example, the expression "adjusted the earnings" will be transformed into "adjust earning".

Once a compound term is recognized in a document or a query, it is added into the document or query. For example, consider a preprocessed text as follows:

```
arm dealer prepare relief supply to soviet union
```

From this segment, we can extract two stored terms "arm dealer" and "soviet union" So the following terms are appended to the original text:

```
arm_dealer soviet_union
```

Once compound terms are extracted from the training texts, the corpus is submitted to the training process of translation models described in Section 2. However, as compounds are considered as units of the texts, the resulting translation models will also contain translations for the compounds, which are usually more accurate than their word-by-word translations.

A. Experiments on CLIR

Table IV shows the CLIR results with both types of translation model. These results are obtained on the same document collection as the one used earlier, but the query set is different.

In these experiments, we separate single words and compound terms into two separate vectors. SMART has the flexibility of building multiple vectors for a document and for a query. Then the global similarity between the document and the query is determined by the weighted sum of the similarities between the vectors. One can assign a relative weight to different vectors of the query to balance their importance in the global similarity.

In our experiments, we tested several values for the relative weights of the single-word vector and the compound-term vector. The above results are obtained with the relative

[5] http://www.cis.upenn.edu/~treebank/home.html

importance of 0.3 to the compound-term vector, and 1 to the single-word vector. This assignment gives the best result.

We can see a great improvement in CLIR effectiveness once the translation model incorporates compound terms, especially for the F-E case. We have not applied the same approach to the Web corpus. However, we could expect similar improvements with the Web corpus when compound terms are incorporated.

TABLE IV.
THE CLIR EFFECTIVENESS WITH DIFFERENT MODELS.

	Word	Compounds (change)
F-E on AP data set	0.1465	0.2591 (+76.86%)
E-F on SDA data set	0.2257	0.2860 (+26.72%)

VII. CONCLUSIONS

In this paper, we described an approach based on parallel texts that has bee used for CLIR at University of Montreal. Globally, our experiments show that the statistical translation models trained on parallel texts are highly useful for CLIR. They can achieve comparable effectiveness to the state-of-the-art approaches. Our further tests with the parallel Web pages mined automatically show that we can arrive at a reasonable level of effectiveness despite the relatively high rate of noise in the training parallel Web pages. This series of experiments show that our method based on parallel Web pages is suitable for CLIR.

Nevertheless, we also observe several aspects that require improvements:

- We encounter problems for translating proper names. Proper names are often treated as unknown words, and are added into the translation as it is. For some names, the spellings in all the European languages are the same, which does not raise particular problems. For some others with different spellings (e.g. “Bérégovoy” in French, but “Beregovoy” in some English documents), this simple approach does not solve the problem.
- Translation by common but non stop- words: Very often, among the top translation words, the common words such as “prendre” and “donner” (“take” and “give” in French) appear with quite strong probability.
- The mined parallel Web pages contain a certain amount of noise. To improve the translation accuracy, a further filtering of noise is necessary.
- We are currently investigating on these problems.

REFERENCES

[1] P. F. Brown, S. A. D. Pietra, V. D. J. Pietra, and R. L. Mercer, The mathematics of machine translation: Parameter estimation. Computational Linguistics, vol. 19, pp. 263-312 (1993).

[2] Buckley, C. (1985) Implementation of the SMART information retrieval system. Cornell University, Tech. report 85-686.

[3] Chen, S. F. Aligning sentences in bilingual corpora using lexical information. Proc. ACL, pp. 9-16, 1993.

[4] J. Chen, J.Y. Nie. Automatic construction of parallel English-Chinese corpus for cross-language information retrieval. Proc. ANLP, pp. 21-28, Seattle (2000).

[5] W. A. Gale, K.W. Church, A program for aligning sentences in bilingual corpora, *Computational Linguistics*, 19: 1, 75-102 (1993).

[6] G. Grefenstette. The Problem of Cross-Language Information Retrieval. In *Cross-language Information Retrieval*. Kluwer Academic Publishers. pages 1-9, 1998

[7] C. Manning, H. Shultze, *Fundamentals of Statistical Natural Language Processing*, MIT Press, 1999

[8] J.Y. Nie, P. Isabelle, M. Simard, R. Durand, Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the Web, *ACM-SIGIR conference*, Berkeley, CA, pp. 74-81(1999).

[9] J.Y. Nie, J.F. Dufort, Combining Words and Compound Terms for Monolingual and Cross-Language Information Retrieval, *Information 2002*, Beijing, July 2002.

[10] Prosise J., Crawling the Web, A guide to robots, spiders, and other shadowy denizens of the Web, PC Magazine - July 1996 (http://www.zdnet.com/pcmag/issues/1513/pcmg0045.htm).

[11] Resnik, Philip (1998) Parallel stands: A preliminary investigation into mining the Web for bilingual text, *AMTA'98*, *Lecture Notes in Artificial Intelligence, 1529,* October.

[12] M. Simard, G. Foster, P. Isabelle, Using Cognates to Align Sentences in Parallel Corpora, *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal (1992).