BITS: A Method for Bilingual Text Search over the Web

Xiaoyi Ma, Mark Y. Liberman
Linguistic Data Consortium
3615 Market St. Suite 200
Philadelphia, PA 19104, USA
{xma,myl}@ldc.upenn.edu

Abstract Parallel corpus are valuable resource for machine translation, multilingual text retrieval, language education and other applications, but for various reasons, its availability is very limited at present. Noticed that the World Wide Web is a potential source to mine parallel text, researchers are making their efforts to explore the Web in order to get a big collection of bitext. This paper presents BITS (Bilingual Internet Text Search), a system which harvests multilingual texts over the World Wide Web with virtually no human intervention. The technique is simple, easy to port to any language pairs, and with high accuracy. The results of the experiments on German – English pair proved that the method is very successful.

1 Introduction

Large parallel corpus was proved to be extremely useful for research in multilingual natural language processing and language teaching, such as statistical machine translation [Brown1990] [Melamed1998], cross-lingual information retrieval [Davis&Dunning1995] [Landauer&Littman1990] [Oard1997], lexical acquisition [Gale&Church1991a] [Melamed1997].
However, due to fees and copyright restrictions, for all but relatively few language pairs, parallel corpora are available only in relatively specialized forms such as United Nations proceedings [LDC], Canadian Parliament debates [LDC], and religious text and software manuals [Resnik&Melamed1997]. The available parallel corpuses are not only in relatively small size, but also unbalanced.
Lack of large parallel corpus makes some research in multilingual natural language processing impossible, for example, the majority of the machine translation researches are rule-based, only a few are statistical machine translation. Some scholars believe that the lack of large parallel corpora makes statistical approach impossible, the researchers don't have large enough parallel corpuses to give some language pairs a shot.
However, the unexplored World Wide Web could be a good resource to find large-size and balanced parallel text. According to the web survey we did in 1997, 1 of 10 de domain websites are German – English bilingual, the number of de domain websites is about 150,000 at that time, so there might be 50,000 German – English bilingual websites in de domain alone. Things are changing since a potential gold mine of parallel text, the World Wide Web, has been discovered. Researchers are making their efforts to mine parallel text from the web [Resnik1998].

This paper presents a method for automatically searching parallel text on the Web. It scans a list of potential websites, finds the bilingual or multilingual websites, downloads them, cleans them up, finds pages which are translation pairs and stores them in a database. The technique is conceptually simple, easy to port to other language pairs. We evaluated the system on German – English language pair, the results indicate that the method is accurate and efficient enough to apply without human intervention. Section 2 lays out the structure of BITS. Section 3 describes the translation pair finder in detail, which is the core of the method. Section 4 presents the experiment results. Section 5 concludes the paper, and Section 6 discusses future work.

2 The BITS Architecture

The BITS architecture is a simple pipeline. Given a particular pair of languages of interest, a candidate generation module generates a list of websites which have a high possibility of being bilingual of the given languages. Then, for each

website on the list, the website language identifier will identify the language property of the website. If it is not a bilingual or multilingual website, then process next website on the list. Otherwise, a web robot downloads all the htmls and plain text files from the website recursively. Afterwards, the htmls are converted to plain text files. Next, a language identifier identifies the language of each text file. Finally, a translation pairs finder finds all the translation pairs and stores them to a database.

2.1 Candidate Websites Generation

To generate a list of candidate websites, we simply find all the websites in the domains which have a high possibility of containing parallel text for the given language pair. According to a web survey we did in 1997, on average, only 1 out of 1000 website is bilingual or multilingual. However, if you focus on some specific domains, you will discover some very interesting fact, for instance, 1 out of 10 website in de domain is German – English bilingual. This is also reasonable for other domains, similarly, ca domain websites are very possible to be French – English bilingual.
Based on this assumption, we can generate the candidate list easily. For example, for German – English, de (Germany), au (Austria) and lu (Luxembourg) domain could be a good start. For each of the candidate domains, a list of all the www servers can be obtained by querying some DNS servers or by crawling the given domain.

2.2 Website Languages Identification

To identify whether a given website is monolingual or multilingual, we look at pages of the top 3 or 4 levels of the website. The language identifier can identify the language of each page. If there are more than one language used in the top 3 or 4 level of a website, we assume the site is at least bilingual. There are cases that a website has pages of two languages but they are not bilingual translations. However, assuming they are bilingual won't hurt.
Given a text file, language identifier tells in which language (natural language) the text is written.
Current language identification techniques include small words technique and N-gram technique [Grefenstette1995]. Either method works well on long sentences (more than 20 words) and that N-gram is most robust for shorter sentences. Both methods are easy to implement. Using short words is slightly more rapid in execution since there are less words than there are N-grams in a given sentence, and each sentence attribute contributes a multiplication to the probability calculation.
In our application, we choose N-gram method. It's almost always true that a lot of web pages contain only very short lists, not sentence, especially for the top level pages. These short lists barely contain short words by which the language identifier used to identify a language, so the short words method fails very often in these cases. N-gram method is, however, still robust in these cases.
The features of our language identifier include:

- Trainable:
  The language identifier could be easily trained on a specified set of languages. For each language, 100K text is needed to train the language identifier.
- Confidence feedback:
  The language identifier should not only give you the language of the text, but also the confidence of the judgement. The reason that this feature is important is that, you can't train the language identifier on all the languages, the confidence gives you a chance to tell whether the language is in the training set. If the confidence is lower than a given threshold, the language is 'unidentified'.

2.3 Website Downloading

We use GNU Wget to retrieve web pages from a remote website.
GNU Wget is a utility designed for retrieving binary documents across the Web, through the use of HTTP and FTP, and saving them to disk. Wget is non-interactive, which means it can work in the background, while the user is not logged in. Analyzing server responses, it distinguishes between correctly and incorrectly retrieved documents, and retries retrieving them as many times as necessary, or until a user-specified limit is reached.
Wget supports a full-featured recursion mechanism, through which you can retrieve large parts of the web, creating local copies of remote directory hierarchies. Wget understands the robot exclusion standard[1] – '/robots.txt', used by server administrators to shield parts of their system from being scanned by web robots. Most of the features of Wget are fully configurable,

[1] See http://info.webcrawler.com/mak/projects/robots/robots.html.

either through command line options, or via the initialization file.
We only retrieve HTML files and plain text files because we are only interested in texts. This makes the retrieval very fast, since in general text files are much smaller than image and audio files.

2.4 HTML Cleanup and Language Identification

The HTMLs are converted to plain text after they are retrieved from remote website. The language of each page is also identified by the language identifier afterwards.
We noticed that very small files decrease the accuracy of language identifier and the performance of translation pairs finder. So, we put a threshold (500 bytes in our experiment) on the plain text files, i.e. if the size of the text file is below the threshold, we throw it away.
This practice doesn't effect the size of our collection a lot, and we get the advantage of more accurate prediction of translation pairs which may benefit further research a lot.

3 Finding Translation Pairs

After the files are cleaned up and language of each page is identified, we end up with two lists of files, one for each language in the language pairs we are interested in, say L1 and L2. The problem remains is how to find translation pairs among the two lists of files.

3.1 Overview

Possible approaches of finding translation pairs include filename and path similarity comparison, file makeup comparison, and content-based similarity comparison. The filename and path similarity approach basically compares the full path (including file name) of a file A in L1 with the full path (including filename) of a file B in L2, if some degree of similarity exists between the full path of A and the full path of B, it's very possible that file A and file B are mutual translations of each other. For example, page http://www.freezone.de/index_d.htm is more likely to be the mutual translation of page http://www.freezone.de/index_e.htm than http://www.freezone.de/news/d_intro.htm, since http://www.freezone.de/index_d.htm is more similar to http://www.freezone.de/index_e.htm than http://www.freezone.de/news/d_intro.htm is. Considering full path as string, the similarity measure of could be of edit distance of two strings, such as the Levenshtein [Levenshtein1965] distance and the Likeit distance [Yianilos1993][Yianilos1997]. The intuition here is that the webmasters tend to name the files with similar names if they talk about the same topic. However, the way a webmaster designs a website could be various, this makes the file name similarity based approach very difficult to give an accurate prediction of translation pairs. And, it happens very often that the files which comment on the same topic could be very much different, since web page designer want to show different viewers different aspects of a topic. This makes things even worse.
The approach based on file makeup comparison assumes that web designers make pages of the same content in two languages the same appearance. This is often true, but still it does not work very well. It filters out pages which are translations to each other but without a similar appearance and accept some pairs which are not mutual translations but with similar makeup. It also fails when HTMLs do not have very much makeup.
Human beings can recognize translations easily because they have at least some degree of knowledge about the languages. The more language knowledge they have, more accurate they can predict. Based on this observation, we propose a content-based approach of finding translation pairs, which understands the languages in some degree.

3.2 Content-based Translation Pairs Finder

If two texts are mutual translations, corresponding regions of one text and its translation will contain word token pairs that are mutual translations. We call these token pairs translational token pairs. For example, in following two sentences, sentence 2 is the German translation of sentence 1:

1. The functionality of the software has been enhanced.
2. Die Funktionalität der Software wurde erweitert.

Word 'functionality' and 'Funktionalität' are translation token pairs, so are 'software' and 'software', 'enhanced' and 'erweitert'.

The following is the algorithm of Translation Pairs Finder.

for each A in L1

```
        tokenize A
        max_sim = 0
        for each B in L2
                Tokenize B
                S = sim(A,B)
                if s > max_sim then
                        max_sim = s
                        most_sim = B
                Endif
        Endfor
        If max_sim > t then
                output (A, B)
        endif
endfor
```

For a given text A in language L1, we first tokenize A and every B in language L2. We measure the similarity between A and every text B in language L2. And then we find the B which is most similar to A, if the similarity between A and B is greater than a given threshold t, then A and B are declared a translation pair.
sim(A, B) is defined as:

$$sim(A,B) = \frac{\text{Number of translation token pairs}}{\text{Number of tokens in text A}}$$

The most straightforward way of finding translation token pairs is using a translation lexicon (each entry of a translation lexicon lists a word in language A and its translation in language B), whenever a pair of words in corresponding region of parallel text is an entry of the translation lexicon, the pair is considered a candidate translation token pair.
For linguistically similar language pairs, such as French and English, candidate translation token pairs can also be found by looking for cognates in corresponding region of parallel text. For example, in the following two sentences:

1. The functionality of the software has been enhanced.
2. Die Funktionalität der Software wurde erweitert.

'functionality' and 'Funktionalität' are cognates, 'software' and 'Software' are cognates, they are considered as translation token pairs.
For language pairs which share lots of cognates, such as French and English, Spanish and English, identifying cognates along will find enough candidate translation token pairs. For other language pairs, such as German and English, Chinese and English, translation lexicons are required.
The cognates approach and translation lexicon approach can be used together to get a better performance.
To find the real translation token pairs among candidates, we use distance-based model of translation equivalence. Thinking of tokens of text A and text B as two coordinates, as illustrated in Figure 1, if the position of a token in text A are too far away from the position of a token in text B, the token are unlikely to be real translation token pair. For example, the pair S in Figure 1. By setting a distance threshold d, we can rule out the false translation token pairs from candidates.

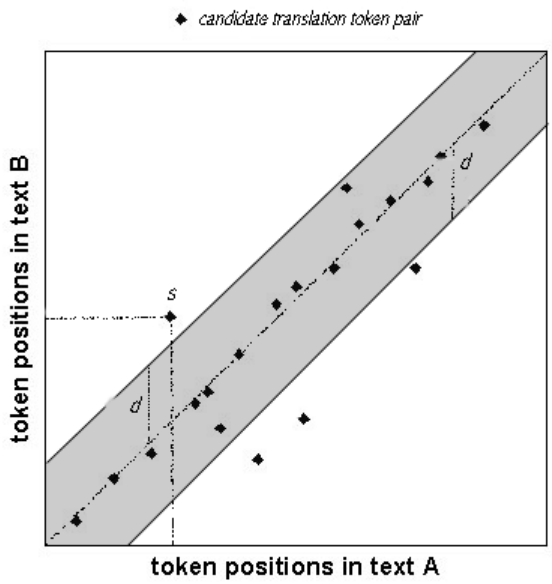

Figure 1. Distance-based model of translation equivalence. Candidate translation token pairs whose co-ordinates lie in the shaded region count as real translation token pairs.

Any translation token pair whose co-ordinate is closer than d would be considered a real translation token pair. The optimal value of threshold d varies with the language pair and the text genre.

To improve the efficiency of the algorithm, before searching for translation token pairs, we compare the size of two files, the number of anchors (something that don't change after being translated, such as numbers, acronyms, usually name of organizations, companies, such as IBM) and number of paragraphs to filter out impossible pairs.

4 Evaluation

The language identifier was trained to recognize 13 languages: English, French, Spanish, German, Italian, Danish, Dutch, Danish, Swedish, Portuguese, Norwegian, Chinese and Japanese. The experiment shows that the language identifier is 100% accurate for text over 500 bytes.

To measure the accuracy of the translation pair finder, we hand picked 300 German pages and 300 English pages from 10 websites, the smallest page is 686 bytes, the largest 32,386 bytes. We found 240 translation pairs manually. Then we ran the translation pair finder on the data. It found 235 translation pairs, 2 of which are wrong. Thus, according to the experiment, its recall and precision are 97.1% and 99.1% respectively.

To measure the feasibility of the method, we ran the experiment on 30,000 .de domain websites. We used a German – English translation lexicon. It has 114,793 entries, 71,726 German words, including inflections. Both German and English stemmer were used in the experiments.
Among 30,000 .de domain websites we picked randomly, 3,415 of them were identified as bilingual or multilingual websites.
Because we're only interested in sentences, so we extracted sentences from each page, and discarded other information, such as lists, tables, and so on. Also, to increase the accuracy of translation pairs finder, we threw away all the pages (contains only sentences) whose size is smaller than 500 bytes.
We ran the experiment on 20 sparc stations during nights. It takes 10 days to complete the task. As a result, among 3,415 bilingual websites, 1,547 of them have more than 1,000 bytes parallel text. The total amount of parallel text we get is 63 Meg bytes.

5 Conclusion

This paper presents the BITS, an automatic system which collects parallel text over the World Wide Web. We conducted several experiments on German-English pair. The experiment results are very encouraging. The method is simple, accurate, easy to port to other language pairs and quite efficient. The method could be a very successful way to collecting parallel text over the Web.

6 Future Work

There are some problems we should work on in the future:

- Balanced web downloading.
- Efficiency of translation pairs finder.
- Html2text conversion. Webmasters designed their homepages in so many different ways, it results in the difficulty to clean them up and keep the sentences from being breaking up. This problem is important because it will affect the accuracy of automatic aligning process.

Acknowledgments

We are grateful to Chris Cieri, Zhibiao Wu, David Graff for useful discussions.

Reference:

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., & Roosin, P. (1990). "A statistical approach to machine translation". Computational Linguistics, 16(2), 79-85.

Church, K.W., & Mercer, R. (1993). "Introduction to the special issue on computational linguistics using large corpora", Computational Linguistics, 19(1), 1-24.

Davis, M., & Dunning, T. (1995). "A TREC evaluation of query translation methods for multi-lingual text retrieval". Fourth Text Retrieval Conference (TREC-4). NIST.

Dunning. T. (1994). "Statistical identification of language". Computing Research Laboratory technical memo MCCS 94-273, New Mexico State University Las Cruces, New Mexico.

Gale, W. A., & Church, K. W. (1991a). "Identifying word correspondences in parallel texts". Fourth DARPA Workshop on Speech and Natural language, Asilomar, California.

Gale, W. A., & Church, K. W. (1991b). "A program for aligning sentences in bilingual corpora". Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, California.

Grefenstette, G. (1995). "Comparing two language identification schemes". Proceedings of the 3rd International Conference on the Statistical Analysis of Textual Data (JADT'95), Rome, Italy.
http://www.rxtc.xerox.com/research/mltt/Tools/guesser.html.

Landauer, T. K., & Littman, M. L. (1990). "Fully automatic cross-language document retrieval using latent semantic indexing". Proceedings of

the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research, pp. Pages 31-38, UW Centre for the New OED and Text Research, Waterloo, Ontario.
LDC (1999). Linguistic Data Consortium (LDC) home page. http://www.ldc.upenn.edu/.
Levenshtein, V. I.(1965). "Binary codes capable of correcting spurisou insertions and deletions of one" (original in Russian). Russian Problemy Peredachi Informatsii, 1:12-25.
Melamed, I. D. (1996). "A geometric approach to mapping bitext correspondence". Conference on Empirical Methods in Natural Language Processing, Philadelphia, Pennsylvania.
Melamed, I. D. (1997). "Automatic discovery of noncompositional compounds in parallel data". Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP-97), Brown University.
Melamed, I. D. (1998). "Word-to-word models of translation equivalence". IRCS technical report #98-08, University of Pennsylvania.
Oard, D. W. (1997). "Cross-language text retrieval research in the USA. Third DELOS Workshop". European Research Consortium for Informatics and Mathematics.
Resnik, P., & Melamed, I. D. (1997). "Semi-automatic acquisition of domain-specific translation lexicons". Fifth Conference on Applied Natural Language Processing, Washington, D.C.
Resnik, P., Olsen, M. B., & Diab, M. (1998). "The Bible as a parallel corpus: Annotating the 'Book of 2000 Tongues'".
Yianilos, Peter (1993). Data structures and algorithms for nearest neighbor search in general metric spaces. Proceedings of the 4th ACM-SIAM Symposium on Discrete Algorithms, pp. 311-321.
Yianilos, Peter (1997). "The Likeit intelligent string comparison facility". Technical Report 97-093, NEC Research Institute.