Mining Parallel Text from the Web based on Sentence Alignment *

Bo Li, Juan Liu and Huili Zhu

School of Computer Science, Wuhan University
Wuhan, 430072, China
whulibo@gmail.com, liujuan@whu.edu.cn

Abstract. The parallel corpus is an important resource in the research field of data-driven natural language processing, but there are only a few parallel corpora publicly available nowadays, mostly due to the high labor force needed to construct this kind of resource. A novel strategy is brought out to automatically fetch parallel text from the web in this paper, which may help to solve the problem of the lack of parallel corpora with high quality. The system we develop first downloads the web pages from certain hosts. Then candidate parallel page pairs are prepared from the page set based on the outer features of the web pages. The candidate page pairs are evaluated in the last step in which the sentences in the candidate web page pairs are extracted and aligned first, and then the similarity of the two web pages is evaluate based on the similarities of the aligned sentences. The experiments towards a multilingual web site show the satisfactory performance of the system.

Keywords: Parallel Corpus, Web Mining, Sentence Alignment, Information Extraction

1. Introduction

Data-driven methods take a more important place in the research field of natural language processing nowadays, which calls an urgent need for enough parallel corpora with high quality. Parallel corpora made up with text in parallel translation are the foundational resource in data-driven natural language processing, which has a direct impact on the effectiveness of this kind of technologies such as statistical machine translation (Brown et al., 1990), cross-lingual information retrieval (Davis and Dunning, 1995; Landauer and Littman, 1990; Oard, 1997) and automatic lexical acquisition (Gale and Church, 1991b; Melamed, 1997). But it is disappointed that there are only a few parallel corpora publicly available today and most of them are small in size, specializing in narrow areas or out-of-date. It will cost too much time and human labor to construct parallel corpora with big scale and high quality, which constraints the increase of parallel corpora. The rapid development of the Internet and the intense intercourse between countries gives hope that we can construct parallel corpora automatically from the web. Through the rough observation, it is found that there are many web sites containing web pages in parallel translation. Based on the above considerations, we present a novel tool called Parallel Web Corpus Construction system (PWCC) to mine parallel corpora automatically from the web, which may help to facilitate the construction of parallel corpora with much less cost.

The PWCC system uses a four-step process to fetch parallel corpora from the web. In the first step, a tool called web spider is employed to fetch all the web pages from specific hosts which

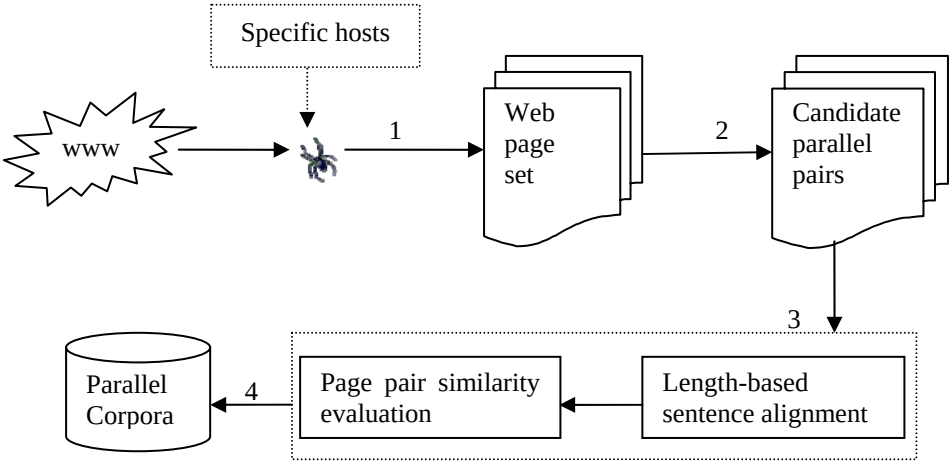
* The work was finished while the first author visited Wuhan Office of Comet Electronics Hong Kong.

probably contain high-quality parallel web pages. The software WebZip[1] is utilized to fetch web pages in the PWCC system. In the second step, candidate parallel web page pairs are prepared from the raw web page set based on the outer features of the web pages. The third step is the key of the whole system, in which the candidate parallel web page pairs produced by the second step are evaluated and the actually parallel pairs are saved. The evaluation module first extracts the sentences from each candidate page pair and aligns them, and then the similarity of the web pages in a pair is evaluated based on the similarities of the sentences which have been aligned. The sentences are aligned based on the length correlation criterion and the sentence similarity is measured by a novel strategy we design. We also design a novel strategy for evaluating the web page similarity based on the aligned sentence similarities. The last step is to save the parallel web pages. It can be concluded from the results of the experiments that the PWCC system is a high-performance and reliable tool for automatically constructing parallel corpora from the web.

The structure of the paper is as follows. The PWCC architecture is introduced in Section 2. The strategy for candidate parallel pair preparation is described in Section 3. The evaluation process is discussed in Section 4 which is the key section of the paper. We practice the experiments and discuss the results in Section 5. The paper is concluded in Section 6.

2. System Architecture

The PWCC system implemented as a four-step process is designed for automatically mining the web for parallel corpora (Figure 1). The first step of the system is the web page fetching process which downloads all the web pages from the hosts specified by the user. In the experiment section, the site of the ministry of foreign affairs of China is selected to be crawled because this site contains a great amount of parallel web pages with high quality. There have already been some free tools to do this work, and the software WebZip is chosen for the PWCC system. The candidate parallel page pairs are prepared in the second step of PWCC. The pairing process mainly relies on the similarity of the URLs, and also some other features such as web page size and anchor text are considered too. The candidate page pairs are then evaluated by the third step



1. Fetch all the web pages from specific hosts and preprocess these pages.
2. Prepare the candidate parallel web page pairs from the web page set.
3. Evaluate the similarity of candidate parallel page pairs based on sentence alignment.
4. Save the web page pairs which are in parallel translation.

Figure 1: The architecture of the PWCC system

[1] http://www.spidersoft.com/

which is also the most important module of PWCC. In this step, the text from the candidate web page pairs is aligned at sentence level first based on the length correlation criterion, and then the similarity of the web pages is measured by combination of the similarities of the sentences that have been aligned. The truly parallel web pages are saved at last.

3. The Candidate Pair Preparation Module

Having been fetched from the web, the web page set consisting both Chinese web pages and English web pages is paired up. Each page is made up of one page from the Chinese page set and one from the English page set. It is important to pair them efficiently and effectively. The number of all the web pages is rather large, so it not only can save much time and RAM space to pair the pages up but also can save a great deal of time for the candidate parallel page pair evaluating step by effective pairing. Moreover, the candidate parallel page pair evaluation step can produce high recall and high precision at a higher probability given the candidate page pairs which are more parallel. There are about two types of methods to pair the web pages up in previous work. One is to pair web pages according to the similarity of the URLs and other features and the other is to simply treat each page from the Chinese page set and each page from the English page set as a candidate pair. It is clear that if the web page set is large, it may be inappropriate to adopt the latter idea because it will produce too many pairs for the evaluation step to process. For the former idea, previous work mostly utilizes the concept of similarity of character strings. They first detect the patterns indicating the language version such as *e*, *eng* and *english* in the URL string according to a pattern list and then use classic algorithms such as the *minimal edit distance* algorithm to measure the similarity of the two URLs. One notable drawback of the existed strategies is that they seldom consider for the deep characters of the URLs and simply treat them as strings. We design a novel candidate page pair preparation algorithm for the problem, which is more powerful than the methods in previous work, and meanwhile its time complexity is considerably low. The algorithm is based on the following assumptions:

Assumption 1: Parallel web pages have the most *common paths* and similar *dictionary depths*.

Assumption 2: Parallel web pages have similar file size.

Assumption 3: Parallel web pages have similar anchor texts.

In *Assumption 1*, the *common paths* mean the common dictionaries the URLs of two web pages share, and the *directory depth* is the layer where the web page lies. For example, we consider two web pages:

Chinese web page P1 URL: *http://www.fmprc.gov.cn/chn/wjdt/zyjh/t263606.htm*

English web page P2 URL: *http://www.fmprc.gov.cn/eng/wjdt/zyjh/t264261.htm*

The number of *common paths* between page *P1* and page *P2* is 2 which are the directories *wjdt* and *zyjh*, and the difference of *directory depths* between page *P1* and page *P2* is 0, because the *directory depth* of page *P1* is 4 and that of page *P2* is 4 too. In *Assumption 2,* the file size is the size of the pure HTML file not including files such as images and audios embedded in the web page. In *Assumption 3,* the anchor text is the text shown at a hyper link, which usually gives the user relevant descriptive or contextual information about the content of the link's destination.

Based on the above assumptions, the strategy we use in the PWCC system is as follows:

For each page *C* in the Chinese page set *C-set,* we find the pages in the English page set *E-set* which can most probably constitute candidate pairs with the page *C.* We select first according to the *common paths* measure. The English pages *Es* which have the most *common paths* with the page *C* are selected first. Then the English pages *Es* are filtered by the *directory depth* criterion. For convenience, the *directory depth* difference between web page *A* and *B* is denoted as *DD(A, B)* and it is clear than *DD(A, B)=DD(B, A).* The *directory depth* filter step is as follow: if two pages *E1* and *E2* in *Es* have different *directory depth* differences with the page *C* and we suppose *DD(E1, C)>DD(E2, C),* we delete *E1* from *Es*. At last, we filter *Es* based on the file size and anchor text criterion. Anchor text is usually a good and brief summary of the content,

so parallel web pages should have similar anchor texts. An easy strategy is designed to evaluate the similarity of the anchor texts in the PWCC system. A bilingual anchor text wordlist which contains words and phrases in parallel translation such as names, place names, duty names, company names and organization names is manually built first. For two anchor texts, if a word or phrase occurs in one anchor text and the bilingual wordlist but its translation in the wordlist doesn't occur in the other anchor text, the two anchor texts are not likely to be the anchor texts of parallel web pages. For each page *E* in *Es,* if the file size of *E* and *C* is too different or the anchor texts of *E* and *C* are not similar, the page *E* is excluded from *Es.* When each page *C* in the Chinese page set *C-set* is processed, we add the corresponding English page set *Es* to the Chinese-English page pair table *Tce.* The whole process is described in the algorithm in *Appendix A.*

For each page *E* in the English page set *E-set,* we perform the similar operation as above and then we can construct another page pair table *Tec.* At last, we can combine *Tce* and *Tec* to get the final candidate parallel page pair set.

4. The Candidate Pair Evaluation Module

The candidate pair evaluation module is the key of the PWCC system. This module consists of a two-step process. The module first aligns the text extracted from the candidate web page pair at sentence level and then evaluates the similarity of the web pages based on the similarities of the aligned sentences. If the similarities of the web page pairs exceed certain threshold, they are treated as parallel wed pages to be saved.

4.1. Sentence Alignment

As the first step of the candidate pair evaluation module, the process of sentence alignment tries to align the sentences extracted from the huge amount of web pages. For the purpose of similarity evaluation in the next step, only coarse alignment is needed here and the alignment process should be designed to work as effectively as possible. There are usually two different strategies for aligning sentences in the parallel text, one is lexicon-based [Wu, 1994] and the other is based on the length correlation of the parallel sentences [Gale and Church, 1991a]. The lexicon-based methods can usually produce aligned sentence pairs with high precision, but the drawbacks of this kind of methods are obvious too. It will cost much time to prepare the fundamental resources such as the bilingual dictionary for the operation, and the operations such as text preprocess and word segmentation are time-consuming too. Compared to the lexicon-based methods, the length-based strategy can achieve a considerably high precision with much less time and space cost. The length-based strategy is used in the PWCC system.

Given a pair of parallel text, the length-based methods would choose the alignment that maximizes the probability over all possible alignments which can be denoted as

$$\arg \max_{A} \Pr(A \mid T_1, T_2) \tag{1}$$

where A is an alignment, and T_1 and T_2 are the English and Chinese text respectively. An alignment A is a set consisting of $L_1\&L_2$ pairs where each L_1 or L_2 is an English or Chinese passage. It is usually hard to calculate from the formula (1), so the approximation of the probability can be made that the probabilities of the individual aligned pairs with an alignment are independent, which can be described as

$$\Pr(A \mid T_1, T_2) \approx \prod_{(L_1 \& L_2) \in A} (L_1 \& L_2 \mid T_1, T_2) \approx \prod_{(L_1 \& L_2) \in A} (L_1 \& L_2 \mid l_1, l_2) \tag{2}$$

where l_1 and l_2 are the lengths of passages L_1 and L_2 respectively. For the English text, the length means the count of the characters in the text. The length of the Chinese text should be

counted as twice of the count of the Chinese characters in the text. Then the problem of maximizing the probability of the alignment given two pieces of text in the formula (2) can be transformed to the minimum problem

$$\begin{aligned}&\arg\max_{A}\Pr(A\mid T_1,T_2)\approx\arg\max_{A}\prod_{(L_1\&L_2)\in A}(L_1\,\&\,L_2\mid l_1,l_2)\\&=\arg\min_{A}\sum_{(L_1\&L_2)\in A}-\log\Pr(L_1\,\&\,L_2\mid l_1,l_2)\end{aligned}\tag{3}$$

The minimization problem in the formula (3) can be implemented in a dynamic programming way.

4.2. Similarity Evaluation

After the above sentence alignment process, the aligned sentences should be utilized to evaluate the similarity of the web pages. It is intuitive that if two pieces of text are in parallel translation, the aligned sentences from them should be roughly in parallel translation too. And the similarity of the sentences is easier to measure than that of the web pages containing the whole text. For example, sentence *S1* in English and sentence *S2* in Chinese are the sentences to be evaluated,

S1: For the first time in three decades Afghanistan is holding parliamentary elections.

S2: 阿富汗 将 举行 三百年 来 的 首次 议会 选举。

The word boundaries of languages such as Chinese and Japanese are not clear, so the process of word segmentation should be practiced on *S2* first. Then we can see that the two sentences *S1* and *S2* have many words in parallel translation such as the pair *Afghanista*n and *阿富汗*. The similarity of the sentences *S1* and *S2* can be measured by the number of words in parallel translation they have. It is clear that a bilingual dictionary is needed here. We have also found that some words such as *in, for* in English and *的, 将* in Chinese have little impact on the meaning of the sentences and sometimes will confuse the similarity evaluation process, so we should remove this kind of words in the evaluation process.

It is supposed that the sentence(s) *SA* in language *A* and the sentence(s) *SB* in language *B* are from a candidate web page pair and have been aligned in the sentence alignment step. The sentences *SA* and *SB* are preprocessed by modules such as word segmentation if needed. Then a strategy is designed to evaluate the similarity of the sentences *SA* and *SB.*

For each word *WA* in the sentence *SA,* if one of its translations in the bilingual dictionary occurs in the sentence *SB,* the *translation count* will be added by 1; else the *translation count* is minus by 0.2. Then the similarity of *SA* and *SB* is given by

$$sim(SA,SB)=\frac{translation\ count}{\max(length(SA),length(SB))}\tag{4}$$

where *sim(SA,SB)* is the *sentence similarity* of the sentence(s) *SA* and the sentence(s) *SB, length(SA)* and *length(SB)* are the count of the words in *SA* and *SB* respectively, the value of the function *max(langth(SA),length(SB))* is the bigger one between *length(SA)* and *length(SB).* The value of the *sentence similarity* is between 0 and 1. The bigger the value is, the more similar the two sentences are.

The strategy for evaluating the *sentence similarity* has been illustrated above, and then we need to evaluate the similarity of the text extracted from the candidate parallel web pages based on the *sentence similarity.* By the sentence alignment process, the candidate parallel text is divided into aligned sentences in two languages. There are many *sentence similarities* for a candidate web page pair, which are denoted as $SS_1, SS_2, SS_3, \ldots, SS_n$. For a candidate parallel page pair, if the similarities of too many aligned sentences are low, we filter this candidate page

pair which is not likely to be parallel directly. Because SS_p and SS_q $(p \neq q)$ are produced independently, the similarity of the candidate web pages can be estimated as

$$similarity = \prod_{i=1}^{n} SS_i \tag{5}$$

In order to facilitate the calculation process, we transform the similarity criterion to its logarithm style which is

$$\log similarity = \sum_{i=1}^{n} \log SS_i \tag{6}$$

The *log similarity* in the formula (6) is a number no more than 0. The threshold for the value in the PWCC system is set to *log 0.65* based on the experiments and our experience.

5. Experiments and Discussions

In this section, experiments are designed to examine the performance of the system. First, two widely used evaluation standards in the information retrieval field are introduced, which are also applicable in our case. In our experiments, *precision* is defined as the proportion of page pairs in parallel translation to the total page pairs that are produced by PWCC. *Recall* is defined as the proportion of page pairs in parallel translation produced by the PWCC system to the total parallel page pairs in the whole web page set.

The factor *number of pairs in parallel translation* must be calculated from the human annotated page pairs, so we invite a native Chinese having learned English for many years to help to annotate these page pairs. To calculate the *recall,* we need to know the total number of parallel pairs in the page set. It is hard to count out the actual number of the parallel pairs in the page set because the web page set is really too large. We build a relatively small test set to test the *recall* of the PWCC system.

The PWCC system first fetches all the web pages from specific hosts. In the experiments, the web spider software WebZip is set to crawl the website of the ministry of foreign affairs of China (http://www.fmprc.gov.cn) which contains a great amount of high-quality Chinese and English web pages in parallel translation. Through the rough observation, most of the web pages in this site are news or governmental archives in both Chinese and English, so the text extracted from the web pages is formal and suitable for auto process by the computer. A web page set containing 40262 Chinese web pages and 17324 English web pages is fetched by the web spider[2]. Some preprocess technologies are practiced on the page set, and then the web pages left are processed by the candidate pair preparation module. After that, the text is extracted from these candidate pairs omitting html tags, advertisements and so on and evaluated by the candidate pair evaluation module. The time it costs is also considerably short. To examine the *precision* of the system, a subset of 400 pairs are selected from all the pairs PWCC produces. A Chinese who has learned English for many years and has a fluent English tongue is asked to annotate the subset, and it is found that 379 pairs in the subset are actually in parallel translation, which accounts for a *precision* of 94.8%. For evaluation of the *recall,* a human-annotated web page pair set is needed here. We construct a web page pair set consisting of 320 parallel page pairs and 80 nonparallel pairs. This web page set is evaluated by the PWCC system and 299 out of the 320 actual parallel pairs are recognized as parallel by the system, which accounts for a *recall* of 93.4%. Both the *precision* and the *recall* show the outstanding performance of the PWCC system. Based on the analysis of the page pairs which are actually parallel but PWCC

[2] The web page set constructed by us has been used in some other experiments before the work in this paper.

considers nonparallel and the page pairs which are actually nonparallel but PWCC considers parallel, we find that most of these pages contain many sentences with similar length, which always brings some problems to the length-based sentence alignment module. But fortunately this kind of web pages only occupy a very small proportion of the whole page set, and will not significantly influence the performance of the system.

6. Conclusion

The PWCC system designed to fetch parallel text from the web is introduced in this paper. The system first uses the web spider to fetch all the web pages from some hosts given by the user, and then candidate parallel web page pairs are prepared by the candidate pair preparation module based on features such as URL and file size. In the last step, candidate parallel web page pairs are evaluated by the sentence alignment-based strategy. We design novel strategies for the second and the third steps, which are then proved to be rather efficient by the experiments. The PWCC system is useful and reliable for automatically constructing parallel corpora from the web.

References

Brown, P. F., J. Cocke, S. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, et al. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics,* 16(2), 79-85.

Davis, M. and T. Dunning. 1995. A TREC evaluation of query translation methods for multi-lingual text retrieval. *Proceedings of The 4th Text Retrieval Conference (TREC).*

Gale, W. A. and K. W. Church. 1991a. A program for aligning sentences in bilingual corpora. *Proceedings* of *The 29th Annual Meeting of the ACL*, Berkeley, CA

Gale, W. A. and K. W. Church. 1991b. Identifying word correspondences in parallel texts. *Proceedings of The 4th DARPA Workshop on Speech and Natural Language.*

Landauer, T. K. and M. L. Littman. 1990. Fully automatic cross-language document retrieval using latent semantic indexing. *Proceedings of The 6th Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research,* UW Centre for the New OED and Text Research, Waterloo, Ontario.

Melamed, I. D. 1997. Automatic discovery of non-compositional compounds in parallel data. *Proceedings of The 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP-97),* Brown University.

Oard, D. W. 1997. Cross-language text retrieval research in the USA. *Proceedings of The 3rd DELOS Workshop.*

Wu, D. 1994. Aligning a parallel English-Chinese corpus statistically with lexical criteria. *Proceedings of The 32nd Annual Meeting of the ACL*, Las Cruces, NM

Appendix A: The Algorithm for Preparing Candidate Parallel Page Pairs

```
for each page C in C-set
{
   maxCommonPath = -1;
   directoryDepthDifference = 100;
   // clear the E-set object
   E-set = null;
   for each page E in E-set
   {
     // CP(E, C) is the count of common paths between page E and C
      if(CP(E, C) > maxCommonPath)
      {
        // LD(E, C) is the file length difference between page E and C
        // AS(E, C) is the similarity of the anchor texts of E and C
         if(LD(E, C) < 20kB && AS(E, C) == true)
         {
```

```
            maxCommonPath = CP(E, C);
           // DD(E, C) is the directory depth difference between page E and C
            directoryDepthDifference = DD(E, C);
            E-set = null;
            Add E to E-set;
          }
       }
       else if(CP(E, C) == maxCommonPath)
       {
          if(DD(E, C) < directoryDepthDifference)
          {
             if(LD(E, C) < 20kB && AS(E, C) == true)
             {
                directoryDepthDifference = DD(E, C);
                E-set = null;
                Add E to E-set;
             }
          }
          else if(DD(E, C) == directoryDepthDifference)
          {
             if(LD(E, C) < 20kB && AS(E, C) == true)
             {
                Add E to E-set;
             }
          }
       }
    }
    // each page in E-Set can constitute a candidate parallel page pair with C
    Add (C, E-set) to Tce;
}
```