Automatic English-Chinese Parallel Corpus Acquisition and Sentences Extraction

Yi LU, Xiong ZHANG, Dequan ZHENG*,

Harbin Institute of Technology, Harbin 150001, China

Abstract

There are lots of valuable resource on Internet which can provide with cross languages and cross areas parallel corpus. Some earlier methods are developed to do this mining work. However, they often use one feature only in the mining process. We use multiple reasonable features of parallel pages to acquire parallel corpus. At last, we also add a SVM classifier which utilize all the features to do the mining work. Surely, it achieve a significant improvement than earlier methods. The evaluation is based on massive manually annotated pairs and our method achieves precision rate of 95% and recall rate of 99%.

Keywords: Parallel Corpus; Sentences Extraction; SVM Classifier

1 Introduction

The construction of parallel corpus is essential to natural language processing, particularly in machine translation which is based on statistical models. Therefore, massive corpus is needed to train the related models.

Years ago, it is difficult to acquire massive corpus. Not only most of corpus are under limited license, but they are specific to a single area. With the development of Internet and more and more sites provide with parallel versions of pages in different languages. In order to utilize Internet resource more efficiently, our paper demonstrate a method that acquire parallel pages from Internet and a method that extract parallel sentences based on the corpus acquired previously.

We use search engine to pinpoint candidate sites which contain parallel text, and crawl them with a web spider. Then a URL similarity module based on some predefined patterns is used to mine candidate pairs. All the pairs are pass to filters based on text length, HTML tags similarity and content-based translation. At last we add a SVM classifier which utilize all the three features mentioned above to do an overall mining and verifying.

Evaluation based on massive test data which consist of 801 pairs shows that our method makes some significant improvement than earlier methods. In overall, we achieve precision rate of 95% and recall rate of 99%.

*Project supported by the National Natural Science Foundation of China (No. 61073130) and the Project of National High Technology Research and Development Program of China (No. 2011AA01A207).

*Corresponding author.

Email address: dqzheng@mtlab.hit.edu.cn (Dequan ZHENG).
The structure of the paper is as follows. In section 2, we introduce some related works which make contribution to this mining work. In section 3, we describe the architecture of our method. In section 4, we add a SVM classifier which utilize all the three features to verify parallel pairs. In section 5, we describe the method we use to extract corpus from verified parallel pairs. In section 6, we use test data to evaluate our method’s overall performance. In section 7, we make some conclusions.

2 Related Works

In recent years, parallel corpus has been much easier to acquire. Some methods have been developed to crawl the parallel web pages from Internet and to build corpus which is used in machine translation and some other corpus-based natural language processing tasks.

Philip Resnik and Noah A. Smith develop the STRAND\[1\] method to mine parallel corpus from Internet. They use a search engine to acquire candidate site first. Then they analyze the pages’ HTML structures and align the tags. The texts between the aligned tags are extracted and are used to build parallel corpus.

Jiang Chen and Jian-Yun Nie developed a method called PTMiner\[2\] to mine data from Internet. First, they submit some requests to search engines to pinpoint the candidate sites. Second, they use a file name filter to acquire candidate parallel pairs. Next, they use some filters based on text length, language and character set, HTML structure and Alignment. At last, the corpus is extracted from the parallel web pages’ text.

The PTI\[3\] method is developed by Jisong Chen, Rowena Chau and Chung-Hsing Yeh. They first download all the data of a host by utilizing a web spider. All the files then pass to filename comparison module. Then some parallel pages are extracted. The remaining pages then pass to a content analysis module. After this two steps, PTI achieve a 93% precision rate and 96% recall rate on 193 pairs.

The WPDE\[4\] is developed by Ying Zhang, Ke.Wu, Jianfeng Gao and P. Vines. They first introduce the ALT information of images to mine candidate sites. Then a file length, file structure and content-translation filters are used to verify candidate pairs. At last, a KNN classifier also improve the overall performance based on the three features mentioned above.

3 Architecture

Our method first use search engines such as Google and Bing to find some potential parallel sites. A site is considered as a parallel site if it has a lot of pages which contain parallel versions in different languages. Some known parallel sites can also be added to the method. For example, due to the history of Hong Kong, almost every website of Hong Kong government consists of simplified Chinese, traditional Chinese and English. Parallel page pairs always have some predefined patterns between different language versions. Then we use some filters based on text length, HTML tags similarity and content translation to verify all the candidate parallel pairs. At Last, lots of corpus are extracted from the verified pairs.
3.1 Pinpoint parallel sites

In this procedure, we can take advantage of the billions of pages which have indexed by search engines. We can construct specific requests to acquire the web sites that we desire. One is to construct “anchor request” which is used widely. Since an anchor text is the description of the hyperlink, it is convinced to treat a site as candidate site when the anchor text contains “English Version” or some other predefined words. For example, we can submit “allinanchor: English version” in Google and analyze the result page afterwards.

The other is to construct “URL pattern request” which is never used by previous methods. Usually, the URL patterns of a bilingual site include “zh-cn” or “en-us” to identify the language of a page. Therefore, we construct a request like “allinurl: zh-cn” in Google to search candidate sites.

There are also lots of parallel sites that we have known such as Hong Kong government’s web sites and some other language learning sites. These sites are valuable resource for the later mining work which should be treated as candidate sites.

Finally, we use a spider to crawl all the text from candidate sites and build directory based hierarchy.

3.2 Extract parallel pairs

We then extract candidate parallel pairs using web page’s URL. A URL consists of a protocol prefix, a host name and some directories infix and a filename suffix with some other language flags. For example, given the URLs of an English-Chinese parallel pair.

We can use a naive substitution method that all the possible infix or suffix are substituted by some other language flags, then we verify whether the corresponding URL exists. Although this method can extract all the possible parallel pairs, it is not efficient on a large scale data processing. Therefore, we turn to an alternative method.

According to observation, there are only a few of ags which specify English languages such as e, en, eng, english and en-us. However, Chinese has much more language ags than English.

Therefore, we use a English hashing and Chinese iteration technique to extract parallel pairs. We hash all the English web page URLs according their filenames. Then we iterate all the
Chinese file, search the corresponding English file. If the generated URL is existed, then a candidate parallel pair is extracted. During this process, there is a possibility that we can not find a corresponding English file. We introduce an URL Edit-Distance evaluation method to evaluate all the files in the English Hashing set which have the same filenames with the Chinese URLs. In this way, we enhance the recall rate of candidate parallel pairs extraction.

In this way, we have extracted 3143 pairs from 10 candidate sites.

3.3 Verifying extracted pairs

After extraction in the previous step, the candidate parallel pairs should be filtered by some filters based on text length, HTML tags similarity and content translation further. In order to achieve a better result, we can specify different thresholds in different filters.

3.3.1 Text length

According to the research of Gale Church[8], equivalent sentences should roughly correspond in length—that is, longer sentences in one language should correspond to longer sentences in the other language. Therefore, we define the function Length() which means the text length of a file. We use the following feature value to represent the text length ratio.

\[ S_{len} = \frac{\text{Length (Chinese)}}{\text{Length (English)}} \]  

3.3.2 HTML tags similarity

Under the same site, bilingual page pairs have highly similarity in tags structure if we remove meta, script, link, span and style. We treat every tag as a single word to calculate the longest common subsequence, which abbreviated as LCS. It can be calculated using a \(O(n^2)\) dynamic programming method. We make some definitions in the following.

\[ L_1 = \text{Length (English)}; \quad L_2 = \text{Length (Chinese)} \]
\[ N_{\text{diff}} = L1 + L2 - 2 \times \text{LCS} \quad (3) \]
\[ N_{\text{all}} = \text{LCS} + N_{\text{diff}} = L1 + L2 - \text{LCS} \quad (4) \]
\[ S_{\text{struct}} = 1 - \frac{N_{\text{diff}}}{N_{\text{all}}} \quad (5) \]

### 3.3.3 Content based translation

Here, we first assume the pair is paralleled, then we use an open-source tool kit called Champollion to extract sentence pairs. It is based on lexical information as well as sentence length. It also provide with multi-sentence to single-sentence alignment ability. We define function \( N_S() \) which returns the number of sentence of a text. At the same time, we define function \( N_A() \) which returns the number of aligned sentence using Champollion Tool Kit. Therefore the content translation feature of a pair can be represented by the following formula.

\[ S_{\text{trans}} = \frac{N_A(\text{en}, \text{cn})^2}{(N_S(\text{en}) \times N_S(\text{cn}))} \quad (6) \]

### 4 Utilize a SVM Classifier

Intuitively, we wonder how much improvement we can acquire if we combine the three features described above.

The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the input, making it a non-probabilistic binary linear classifier. We use these three features as characteristic vectors to train our SVM[5] classifier. Then we test the trained classifier with test set.

### 5 Extract Parallel Sentence

#### 5.1 Methods and tools

Currently there are a number of automatic sentence alignment approaches can be utilized, some are pure length based, some are lexicon based, and some are mixture of length and lexicon. To align close language pairs, like alphabetic Indo-European languages, length based approaches are able to achieve a fairly high precision([8]). However, the pure length method may confront problems when it is applied to align remote language pairs, like English and Chinese. In this kind of situation, using an aligner which mixes length and lexicon method will have a better performance. So this paper decided to choose an open source robust parallel text sentence aligner Champollion as our align tool.

Champollion use a similarity function to portrait the resemblance of two segments, E and C. E and C are sets of tokens like these:

\[ E = e_1; e_2; \ldots; e_{m-1}; e_m \quad (7) \]
\[ C = c_1; c_2; \ldots; c_{n-1}; c_n \quad (8) \]

where \( e_i \) and \( c_j \) are word tokens.
And the idea of tf-idf weight which is often used in Information Retrieval is borrowed to define \( stf \), the number of occurrences of a term within a segment, and \( idtf \), which can be computed by

\[
    idtf = \frac{T}{\text{occurrences in the document}}
\]

where \( T \) is the total number of terms in the document.

Then \( k \) is defined as the number of word pairs that can be recognized between the two segments. And the similarity of \( E \) and \( C \) is:

\[
    \text{sim}(E, C) = \sum_{i=1}^{k} \lg(stf(e'_i) \times idtf(e'_i) \times \text{alignment}_{i,j} \times \text{length}_{i,j}(E, C))
\]

In Champollion the alignment penalty is 1 for 1-1 alignment and a number between 0 and 1 for other kinds of alignments, length penalty is a function of the length of the source segment and the length of the target segment.

Then a dynamic programming algorithm is used to find the alignment which has the maximum similarity. This process is just like [8], except Champollion allows 1-0, 0-1, 1-1, 2-1, 1-2, 1-3, 3-1, 1-4 and 4-1 alignment.

The recursion function of the dynamic programming process is:

\[
    S(i; j) = \begin{cases} 
        S(i - 1, j) + \text{sim}(\text{Seg}_{i,1}, \phi) \\
        S(i, j - 1) + \text{sim}(\phi, \text{Seg}_{j,1}) \\
        S(i - 1, j - 1) + \text{sim}(\text{Seg}_{i,1}, \text{Seg}_{j,1}) \\
        S(i - 1, j - 2) + \text{sim}(\text{Seg}_{i,1}, \text{Seg}_{j,2}) \\
        S(i - 2, j - 1) + \text{sim}(\text{Seg}_{i,2}, \text{Seg}_{j,1}) \\
        S(i - 1, j - 3) + \text{sim}(\phi, \text{Seg}_{j,3}) \\
        S(i - 2, j - 2) + \text{sim}(\text{Seg}_{i,2}, \text{Seg}_{j,2}) \\
        S(i - 1, j - 4) + \text{sim}(\phi, \text{Seg}_{j,4}) \\
        S(i - 3, j - 1) + \text{sim}(\text{Seg}_{i,3}, \text{Seg}_{j,1}) \\
        S(i - 2, j - 3) + \text{sim}(\text{Seg}_{i,3}, \text{Seg}_{j,2}) \\
        S(i - 1, j - 5) + \text{sim}(\phi, \text{Seg}_{j,5}) \\
        S(i - 3, j - 2) + \text{sim}(\text{Seg}_{i,3}, \text{Seg}_{j,2}) \\
        S(i - 2, j - 4) + \text{sim}(\text{Seg}_{i,2}, \text{Seg}_{j,3}) \\
        S(i - 3, j - 3) + \text{sim}(\phi, \text{Seg}_{j,4}) \\
        S(i - 4, j - 1) + \text{sim}(\text{Seg}_{i,4}, \text{Seg}_{j,1}) \\
    \end{cases}
\]

where \( \text{Seg}_{a,b} \) are all segments numbered between \( a \) and \( b \), inclusively.

### 5.2 Sentence extraction

To extract parallel sentence from the corpus extracted from the website, a format process to the raw data is needed to satisfy the input requirement of Champollion. Because of the huge amount of the raw data, A Python script is used to help us to accelerate the format process. At first this paper just use some simple punctuation, like period in English and Chinese or the question mark, to split the whole text into sentences. After that we rewrite the sentences into a new file in the required format and run the Champollion tool to align sentences.

After improving the split method, using extra rules to change the text into more accurate sentences. Then again we run the Champollion tool to align sentences. With the more accurate sentence split we’ve got a better precision and recall.
Table 1: Results of three classifiers

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text length ($0.7 \leq S_{len} \leq 1.5$)</td>
<td>94.70%</td>
<td>87.11%</td>
</tr>
<tr>
<td>HTML tags similarity ($S_{struct} \geq 0.83$)</td>
<td>94.59%</td>
<td>97.71%</td>
</tr>
<tr>
<td>Content translation ($S_{trans} \geq 0.5$)</td>
<td>94.87%</td>
<td>95.41%</td>
</tr>
</tbody>
</table>

Table 2: Results of intersection and union

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>intersection</td>
<td>97.42%</td>
<td>81.09%</td>
</tr>
<tr>
<td>union</td>
<td>92.21%</td>
<td>99.79%</td>
</tr>
</tbody>
</table>

6 Experiments and Results

In this section, we describe the test data we use to evaluate our method and the experimental result.

We randomly select a website that we crawled previously and use its web pages as our test data. There are 801 candidate pairs extracted in all. We annotated 698 parallel pairs manually and use these pairs as our test set. The result of three classifiers is illustrated in table 1.

Then, we intersect and union the three filters directly. The result is illustrated in 2.

Later, we divide the test data into three groups and do the test with our SVM classifier with a 3-cross validation[6]. Table 3 illustrates the results.

In overall, three basic classifiers can achieve a satisfying result. At the same time, using the fusion of three features, a little improvement is acquired based on a 3-cross validation. We achieve an average of 95% precision rate and an average of 99% recall rate. The recall rate is greatly improved.

Table 3: Results of the SVM classifiers

<table>
<thead>
<tr>
<th>Train set</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>94.49%</td>
<td>99.57%</td>
</tr>
<tr>
<td>#2</td>
<td>95.86%</td>
<td>99.57%</td>
</tr>
<tr>
<td>#3</td>
<td>93.56%</td>
<td>99.79%</td>
</tr>
</tbody>
</table>

After we get the parallel corpus, we run our align programs, including sentence splitting and aligning. Totally we’ve extracted 16252 parallel sentences from 801 parallel web pages. We’ve noticed that the precision and recall of the result are partly depended on the accuracy of the quality of the sentence split process. Using a basic split method we’ve got a precision of 87.3%, with a recall of 97.2%, while using a enhanced split method we’ve got a precision of 90%, with a recall of 98.7%.
7 Conclusions

This paper describes a novel method for bilingual corpus acquisition. We introduce a URL searching technique to mine candidate parallel sites and a SVM classifier to verify candidate parallel pairs. Our method can achieve a 95% precision rate and a 99% recall rate which is a relative significant improvement than previous works.

Acknowledgement

This work is supported by the national natural science foundation of China (No. 61073130) and the project of National High Technology Research and Development Program of China (863 Program) (No. 2011AA01A207).

References