A Novel Method for Bilingual Web Page Acquisition from Search Engine Web Records

Yanhui Feng, Yu Hong, Zhenxiang Yan, Jianmin Yao, Qiaoming Zhu
School of Computer Science & Technology, Soochow University
{20094227002, hongy, 20074227065071, jyao, qmzhu}@suda.edu.cn

Abstract

A new approach has been developed for acquiring bilingual web pages from the result pages of search engines, which is composed of two challenging tasks. The first task is to detect web records embedded in the result pages automatically via a clustering method of a sample page. Identifying these useful records through the clustering method allows the generation of highly effective features for the next task which is high-quality bilingual web page acquisition. The task of high-quality bilingual web page acquisition is a classification problem. One advantage of our approach is that it is search engine and domain independent. The test is based on 2516 records extracted from six search engines automatically and annotated manually, which gets a high precision of 81.3% and a recall of 94.93%. The experimental results indicate that our approach is very effective.

1 Introduction

There have been extensive studies on parallel resource extraction from parallel monolingual web pages of some bilingual web sites (Chen and Nie, 2000; Resnik and Smith, 2003; Zhang et al., 2006; Shi et al., 2006). Candidate parallel web pages are acquired by making use of URL strings or HTML tags, then the translation equivalence of the candidate pairs are verified via content-based features.

However, we observe that bilingual resources may exist not only in two parallel monolingual web pages, but also in single bilingual web pages. For example, many news web pages and English learning pages are bilingual. Based on this observation, researchers have proposed methods to improve parallel sentences extraction within a bilingual web page. Jiang (2009) uses an adaptive pattern-based method to mine interesting bilingual data based on the observation that bilingual data usually appears collectively following similar patterns. Because the World Wide Web is composed of billions of pages, it is a challenging task to locate valuable bilingual pages.

To acquire bilingual web pages automatically, a novel and effective method is proposed in this paper by making use of search engines, such as Baidu (http://www.baidu.com). By submitting parallel sentence pairs to the given search engine, lots of result pages with web records are returned, most of which are linked to bilingual web pages. We first identify and extract all result records automatically by selecting and analyzing a sample page with a clustering method, and then select high-quality bilingual web pages from candidates with classification algorithms.

Our method has the following advantages:

1. Former researchers extract parallel corpus from specific bilingual web sites. Since search engines index amounts of web pages, and we aim to acquire bilingual pages based on them, our method expands the corpus source greatly.

2. For one search engine, only one sample result page is used to generate the record wrapper. Then the wrapper is used to identify web records from other result pages of the same search engine. Compared with existing data record extraction technologies, such as MDR (Liu et al., 2003; Zhai and Liu, 2006), our method is more effective and efficient.

Coling 2010: Poster Volume, pages 294–302,
Beijing, August 2010

3. We model the issue of verification bilingual pages as a binary-class classification problem. The records acquired automatically and annotated manually are utilized to train and test the classifier. This work is domain and search engine independent. That is to say, the records acquired from any search engine in any domain are used indiscriminately as training and testing dataset.

The rest of the paper is organized as follows. Related works are introduced in section 2. Section 3 provides an overview of our solution. The work about bilingual page acquisition and verification is introduced in section 4 and 5. Section 6 presents the experiments and results. Finally section 7 concludes the paper.

2 Related Work

As far as we know, there is no publication available on acquiring bilingual web pages. Most existing studies, such as Nie (1999), Resnik and Smith (2003) and Shi (2006), mine parallel web documents within bilingual web sites first and then extract bilingual sentences from mined parallel documents using sentence alignment method.

In this paper, the candidate bilingual web pages are acquired by analyzing web records embedded in the search engines' result pages. Therefore, record extraction from result pages is a critical technique in our method. Many researches, such as Laender (2002), have been developed various solutions in web information extraction from kinds of perspectives.

Earlier web information extraction systems (Baumgartner et al., 2001; Liu et al., 2000; Zhai and Liu, 2005) require users to provide labeled data so that the extraction rules could be learned. Yet such semi-automatic methods are not scalable enough to the whole Web which changes at any time. That's why more and more researchers focus on fully or nearly fully automatic solutions.

Structured data objects are normally database records retrieved from underlying web databases and displayed on the web pages with some fixed templates, so automatic extraction methods try to find such patterns and use them to extract more data. Several approaches have succeeded to address the problem automatically without human assistance. IEPAD (Chang and Lui, 2001) identifies sub-strings that appear many times in a document. By traversing the DOM tree of the Web page, MDR extracts the data-rich sub-tree indirectly by detecting the existence of multiple similar generalized-nodes. The key limitation is its greedy manner of identifying a data region. DEPTA (Zhai and Liu, 2005) uses visual information (locations on the screen at which the tags are rendered) to infer the structural relationship among tags and to construct a tag tree. NET (Liu and Zhai, 2005) extracts flat or nested data records by post-order or pre-order traversal of the tag tree. ViNTs (Zhao et al., 2005) considers the web page as a tag tree, and utilizes both visual content features as well as tag tree structures. It assumes that data records are located in a minimum data-rich sub-tree and separated by separators of tag forests. Zhao (2006) explicitly aims at extracting all dynamic sections from web pages, and extracting records in each section, whereas ViNTs focuses on record extraction from a single section. Miao (2009) figures out how tag paths format the whole page. Compared with the previous method, it compares pairs of tag path occurrence patterns to estimate how likely these tag paths represent the same list of objects instead of comparing one pair of individual sub-trees in the record. It brings some noise. We follow this method and make appropriate improvement for our task.

3 Basic Concepts and Overview

3.1 Basic Concepts

Some basic concepts are introduced below.

Figure 1. An example of search engine return

Tag Path: The path of a tag consists of all nodes from the tree root <html> to itself. We use tag path to specify the location of the tag. The tag paths are classified into two types: text tag paths and non-text tag paths.

Data Record: When a page is considered as strings of tokens, data records are enwrapped by one or more tag paths, which compose the visually repeating pattern in a page. This paper aims to extract such structured data records that are produced by computer programs following some fixed templates, while whose contents are usually retrieved from backend databases. For example, there are four records in Figure 1.

3.2 Method Overview

We can get much more bilingual web pages by submitting parallel sentence pairs to the search engine than submitting monolingual queries. Based on this observation, our work is as shown in Figure 2. The algorithm consists of two steps: 1) Record wrapper generation. By submitting parallel sentence pairs to search engines, result pages containing lots of web records are returned. In order to generate record wrappers, we select and analyze a sample page and then apply clustering method to tag paths with similar patterns. We apply these wrappers to extract more records, which are linked to candidate bilingual web pages. 2) High-quality bilingual page acquisition. In order to acquire high-quality bilingual pages from candidates, a binary classifier is constructed to decide whether the candidate pages are bilingual or not. In order to improve the classifier, some useful resources are used, such as a dictionary and translation equivalents.

However, a result page often contains some information irrelevant to the query, such as information related to the hosting site of the search engine, which increases the difficulty of record extraction. Besides, there are also many irrelevant records irrelevant to the query. So our focus is to acquire plenty of features to filter out the irrelevant pages from the candidates.

In this paper, the first result page is chosen as the sample page and Affinity Propagation (AP) clustering is used. The reason lies in Frey and Dueck (2007), which proves that to produce the groups of tag paths; the AP algorithm does not require the three restrictions: 1) the samples must be of a specific kind, 2) the similarity values must be in a specific range, and 3) the similarity matrix must be symmetric. In order to decide the type of a page, the Support Vector Machines (SVM) (Cortes and Vapnik, 1995) classifier on Fuzzy C-means is constructed combining with word-overlap, length and frequency measures. SVM is well-fitted to treat such classification problems that involve interrelated features likes ours, while most probabilistic classifiers, such as Naïve Bayes classifier, strongly assume feature independence (DuVerle and Prendinger, 2009).

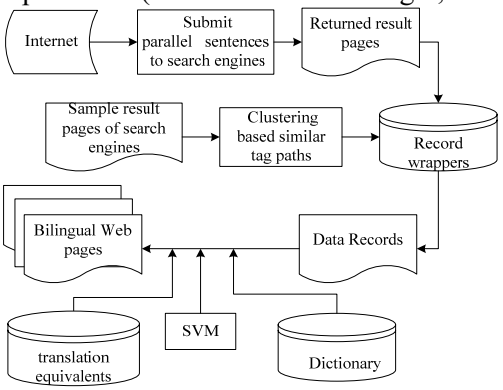

Figure 2. Overview of the method

4 Bilingual Page Acquisition

4.1 Result Page Extraction

The result pages of a search engine consist of a ranked list of document summaries linked to the actual documents or web pages. A web document summary typically contains the title and URL of the web page, links to live and cached versions of the page and, most importantly, a short text summary, or a snippet, to convey the contents of the page. Such snippets embedded in result pages of search engines are query-dependent summaries. White (2001) finds the result pages are sensitive to the content and language of the query. If the query is monolingual, the returned search results are mostly monolingual, while the result pages are bilingual if the query is bilingual. In order to acquire more bilingual web pages, we submit parallel translation pairs. Figure 1 gives an example result page from Baidu, in which the snapshot consists of four records related to the query, which consists of "I see." and its translation "我明白了。". The results have

more effective advantages than submitting the query "I see." or "我明白了。" respectively.

4.2 Clustering With Path Similarity

Given a web page, we get the occurrence positions of each tag path the same as the sequence in the preorder traversal of the page's DOM tree. Certainly, there are many tag paths which appear several times in the whole page. So an inverted mapping from HTML tag paths to their positions is built easily. For example, there are 599 tag paths formatting the sample page in Figure 1, and after the inverted mapping, we acquire 86 unique tag paths in all. Only tick off one part of the results as shown in Table 1, where P_i represents the *ith* unique tag path, and the vector S_i is defined to store the occurrence positions of P_i in the third column.

As introduced above, detecting visually repeating tag paths is a clustering problem. Above all, a factor in determining the clustering performance is the choice of similarity functions, which captures the likelihood that two data samples belong to the same cluster. In our case, the similarity scores between two tag paths aim to capture how their positions are close to each other and how they interleave each other.

With the purpose of characterizing how close two tag paths appear, we only acquire the distance between paths' average positions, which is easy to obtain by the acquired occurrence vectors. For example, the average position of P_{11} and P_{15} in Table 1 is 227 and 215, so the distance between them is 12.

i	Unique Tag Path (P_i)	Occurrences (S_i) of P_i
1	\html	1
3	\html\head\#text	3,4,7,8,9
9	\html\body\table	84,93,115,146,180, 217,258,292,335,372, 406,437
11	\html\body\table\tr	15,85,94,116,147,181, 218,259,293,336,373, 407,438
14	\html\body\table\tr \td\#text	18,21,24,27,55,79,87, 91,97,111,113
15	\html\body\table\tr \td\a	19,88,118,149,183, 220,261,295,338,375, 409,440

Table 1. Unique tag paths of the sample page

However, the most difficult problem is how to capture the interleaving characteristic between two tag paths. Before doing that, another vector O_i is produced. $O_i(k)$ indicates whether the tag path P_i occurs in the position k or not by its value. In addition, the value is binary that 0 or 1, and 0 shows P_i doesn't occur in the position k, while 1 shows the opposite. Of particular note, the length of each O_i is equal to the total number of HTML tags that formatting the whole web page. Take the tag path P_3 ("\html\head\#text") in Table 1 as an example, whose position vector O_3 is (0, 0, 1, 1, 0, 0, 1, 1, 1, 0… 0), and the vector's length is 599, because there are totally 599 tag paths formatting the sample page in Figure 1.

Based on the position vectors, we capture how tag path P_i and P_j interleave each other by a segment D_{O_i/O_j} of O_i divided by O_j. We aim to find such tag paths that divide each other in average. In other words if the variance of counts in the segment D_{O_i/O_j} is stable, they are likely to be grouped in the same cluster. So, we define the interleaving measure μ in terms of the variances of D_{O_i/O_j} and D_{O_j/O_i} as:

$$\mu(O_i,O_j)=\max\{Var(D_{O_i/O_j}),Var(D_{O_j/O_i})\} \quad (1)$$

where D_{O_i/O_j} is acquired by O_j as follows: if value of $O_j(k)$ is 1, $O_i(k)$ is a separator to segment itself into several regions. The value of every element in the segment is the count of P_i that occurs in every region, which is the number of 1 in the region.

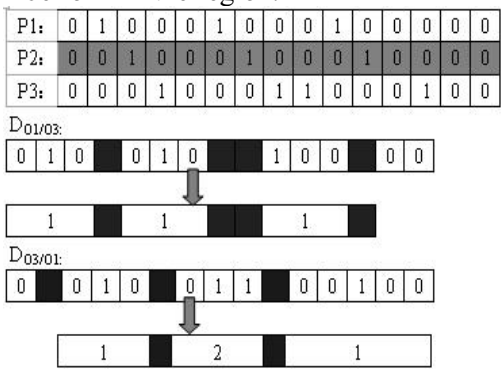

Figure 3. An Example of tag paths

In addition, there may be many consecutive separators in O_i, and we integrate them into one. Besides, the segment is a non-empty set. So if there is no occurrence of P_i in one region, we

will ignore this special region. Figure 3 shows three tag paths. P_1 and P_2 are likely to belong to the same cluster because of their regular occurrences, whereas the occurrences of P_3 are comparatively irregular. By our method, D_{O_1/O_3} = {1, 1, 1} and D_{O_3/O_1} = {1, 2, 1}. We integrate separators once and ignore an empty region in the process of getting D_{O_1/O_3}.

Both the score of the closeness measure and the interleaving measure for any two tag paths are non-negative real numbers. And a smaller value of either measure indicates a high frequency that the two tag paths appear regularly. The measure $\sigma(P_i, P_j)$ defined below is inversely proportional of these two measures.

$$\sigma(P_i, P_j) = \frac{\varepsilon}{c(S_i, S_j) \times \mu(O_i, O_j) + \varepsilon} \quad (2)$$

where ε is a non-negative term that avoids dividing by 0 and normalizes the similarity value so that it falls into the range (0, 1]. In our experiment, we choose ε = 10. By Equation 2, we calculate the similarity value of any pair of tag paths. As expected, the pairwise similarity matrix is fed into the AP clustering algorithm directly, and each cluster acquired from AP clustering contains *n* tag paths, which indicates that those *n* paths appear repeatedly together with high frequency, and the tag paths that have no remarkable relation are spilt into different clusters. For the given sample page in Figure 1, the number of identified clusters is 16.

We observe that HTML code of most data records contain more than three HTML tags, so we only examine the clusters containing four or more visual signals. In the clustering result of sample page in Figure 1, there are three clusters' sizes less than four. Meanwhile, we also note that:

1. The feature page of a common search engine usually contains 10 or more web records with similar layout pattern. So we define a threshold T=3. If an ancestor tag path doesn't occur more than T times, we believe these tag path dose not lead a record.
2. Usually the content of the result pages returned by search engines is completely related to the queries, which means the data records that we are interested in are distributed in the whole page as main component. So the occurrence position of valuable tag paths must be global optimization. In this paper, the scope between beginning and ending occurrence must be wider than three quarters of the length of the web page.

Thus, we get essential clusters fit with above observations, which is denoted by C= {C_1, C_2…C_M}. Once we have the essential clusters, we apply them in new web page of the same search engine to identify data records.

4.3 Data Record Extraction

Based on the essential clusters, we extract the exact data records from the real content of text tag path that follow the ancestor tag path.

In order to describe the extraction process in details, we firstly define D_{aI} as the child tag paths of an ancestor tag path P_a, and suppose that (Pos_1… Pos_i… Pos_m) is the occurrence vector of P_a, which means at each position Pos_i the tag path P_i occurs. $D_{a(i)}$ is such a tag path set that the position Pos of every path in it is $Pos_i < Pos < Pos_{i+1}$. In the meantime, such path strings must begin with the same prefix of P_a. Such as in Table 2, $D_{a\ (i)}$ contains tag paths from Pos_i to Pos_{i+1}-1, and we obtain the *ith* records embedded in the result pages by acquiring the real content of all text tag paths in $D_{a(i)}$.

Occurrence of P_a	D_{aI} of P_a	Child tag path
Pos_1		P_a:\html\body\table\tr
Pos_1+1	$D_{a(1)}$	P_t:\html\body\table\tr\…
……		……
Pos_2-1		P_k: ……
⋮	⋮	⋮
Pos_i		P_a:\html\body\table\tr
Pos_{i+1}	$D_{a(i)}$	P_t:\html\body\table\tr\…
……		……
Pos_{i+1}-1		P_n: ……
⋮	⋮	⋮
Pos_m	$D_{a(m)}$	P_a:\html\body\table\tr
……		……

Table 2. Collection of child tag paths for ancestor tag path

5 Bilingual Web Page Verification

Based on the previous work, we capture a list of records based on a holistic analysis of a result page, and each record contains snippets and URLs related to the query. In this section, we aim to decide whether the candidate pages that returned records are linked to are bilingual or not by putting some statistical features (collected from snippets) into an effective SVM classification.

To the acquired snippets, some necessary preprocessing is made before we acquire useful features. We remove most of the noise that affect the precision and robustness of the entire system by such methods as recovery of abbreviation words, deletion of noisy words, amendment for half or full punctuations and simplified or traditional characters, and so on.

The snippet is described with more regular contents after preprocessing. We cut the snippet into several segments by its language. Each segment of the snippet is just represented in one language, which is either English or Chinese in this paper and different from its adjacent segments. So the source snippets are transferred into such language strings that consist of C and E, where C stands for Chinese and E stands for English. It is unlikely that continuous C or E exists in the same language string. We store the real text T_c (T_e) that each C (E) stands for. We take the snippet "I see. 我明白了。I quit! 我不干了！" as example, its language string is "ECEC" and real text string is $T_eT_cT_eT_c$, where the two T_e stand for "I see" and "I quit", the two T_c stand for "我明白了" and "我不干了".

Note that different feature functions for the classifier will lead to different results, it is important to choose feature functions that will help to discriminate different classes. In this paper, the SVM classifier involves word-overlap, length and frequency features. We define these three features based on the snippet itself as follows:

(1) Word-Overlap measure

Word overlap judges the similarity of Chinese term and English term. In this paper, we acquire the word-overlap score between any two adjacent language segments. The similarity *Score(c_res,e_res)* of Chinese term and English term is based on word-overlap as following:

$$Score(c\_res, e\_res) = \frac{\sum_{i=1}^{p} Max_{1\le j\le q}(Sim(c_i, e_j))}{\phi} \quad (3)$$

where the denominator is normalization factor, and in our experiment we select $p+q$ as its value, where p stands for the length of Chinese term and q stands for the length of English term. In addition, c_i stands for the *ith* word of Chinese term and e_j stands for the *jth* word of English term. $Sim(c_i,e_j)$ in Liu (2003) and Deng (2004) stands for the similarity of Chinese word c_i and English word e_j.

In our experiment, the Chinese and English sub-snippets are equivalent to Chinese and English sentences of the bilingual pages. In the segmented snippet, with regard to each sub-snippet *T*, which is at even position in the language string, we separately evaluate the intermediate score for snippet *T* with its left and right neighbors by Equation 3. Especially when *T* doesn't have right or left neighbor, the score for *T* with its null neighbor is 0. So for every sub-snippet that needs to be scored the word-overlap score, there are two candidate scores with its adjacent neighbors. Then we choose the higher value as one item of an intermediate result vector. Either the length of the language string is $2 \times n$ or $2 \times n+1$, the length of intermediate vector is n, and the final score is computed as follows:

$$Score(s) = \frac{\sum_{k=1}^{n} InV_k}{n \times m} \quad (4)$$

where *Score(s)* stands for the final score of snippet s on the word-overlap measure, and vector *InV* is the intermediate result vector as mentioned before. The length of the vector *InV* is n, and m is the number of its items that is not equal to zero. *m/n* is used as a useful measure of length, because it indicates how many parallel pairs are there in the same snippet.

(2) Length-Based measure

We acquire three scores about length measure. Take the language string "ECECEC" as example, we use "$E_1C_1E_2C_2E_3C_3$" to replace it for simple description. We acquire one score of the length measure as follows:

$$Score\ (s) = \frac{\sum_{i=1}^{m} (Len\ (c) + Len\ (e))}{Len\ (s)} \quad (5)$$

where *s* and *m* stand for the same as in Equation 4. In addition, *c* and *e* stands for such sub-snippet that *Score(c,e)* contributes to $\sum_{k=1}^{n} InV_k$. The function *Len(s)* is to compute the number of words in the sentence.

We acquire the length of language string. If the length is too long or too short, the associated web page is unlikely to be a bilingual page. At the same time, we are not interested in some language strings although the lengths of them are appropriate. So we also store the variances of lengths about each sub-snippet.

(3) Frequency-Based measure

According to the result pages, queries often occur in the title, snippet, or advertisements. They are highlighted to make them easier to identify. Hence we aim to acquire the frequency of the query in one whole snippet as a feature.

Based on the three measures above, a number of records (containing snippets and URLs) for training and testing can be converted them into a 6-dimensional feature space. In our experiments, nonlinear SVM with Gaussian Radial Basis Function (RBF) kernel is used. The performance of the SVM classifier indicates that it is a reliable way to verify whether the page is bilingual or not by the content of snippet.

6 Experiments and Results

6.1 The Data Set

To acquire enough experimental data, we collect from Google, Baidu, Yahoo, Youdao, Bing and Tecent Soso, and the effectiveness of our algorithm is evaluated based on the data set from these six search engines.

Result records of search engines are collected by program and by human beings with submitting different queries respectively. They are used for checking the performance of record extraction. When evaluating the method of verification bilingual web pages, 2300 records (60% are positive instances) are chosen for training the SVM classifier, and other 230 are selected randomly as test records from the whole record set.

The training data is annotated by human in two methods. The first method is motivated by the content of each source snippet. The annotators assign the type of web pages by scanning the text of every snippet. If the snippet contains many parallel term pairs, we annotate the page as bilingual or monolingual if not parallel. We also use another annotation method, which is to reach the URL by the Internet Explorer. By checking the content of the real web page, annotators decide the type of the candidate pages. And the biggest difference between the two public hand-classified dataset appears when some snippets of candidate pages have no clues in their content to predict classifications.

6.2 Evaluation On Bilingual Page Acquisition

The entire system is evaluated by measuring the performance of the binary SVM classifier. And how the classifier performance changes with three features is shown in Table 3, where W, L and F separately stand for the word-overlap, length and frequency measures.

In order to improve the performance of word-overlap measure, we use not only the bilingual dictionary but also translation equivalents, which are extracted from parallel corpora. Because the bilingual dictionary doesn't contain all necessary entries, the classifier with only word-overlap measure accepts many wrong pairs.

Feature	W	W +L	W +L+F
Precision	70.2%	81.02%	85.10%

Table 3. SVM Classifier Performance changes with more features added to the classifier

Table 3 shows that the length feature and the frequency feature have a significant effect on bilingual web page verification because of the natural relationship among queries, snippets and true web pages.

N	#1		#2	
	P(%)	R(%)	P(%)	R(%)
1	85.1	92.3	75%	84.8
2	80.7	95.1	72.8	85.7
3	78.1	97.4	71.0	93.0
aver	81.3	94.93	72.93	87.83

Table 4. Performance versus training data types

Three experiments of verification bilingual web pages based on two different training datasets are conducted whose results are shown in Table 4. #1 stands for the data set annotated by snippets, and #2 stands for the training data annotated by URLs. Precision and recall are used to evaluate our method. The average precision based on training dataset #2 is 73%, which is lower than the precision of 81.3% resulting from the dataset #1, because in many cases, some snippets are weakly related with real text in the real pages introduced by search engine summarization algorithm. From the table, we also see that the recalls in dataset #1 and #2 are both relatively high, which means our classifier can select high-quality bilingual pages with high accuracy.

6.3 Evaluation On Web Record Extraction

Record extraction has significant effect on bilingual web page collection. A useful intermediate evaluation of the whole scheme is conducted by measuring the performance of record extraction.

We built a prototype system to test the algorithm of record extraction based on the clustering of similar records. On a laptop with a Pentium M 1.7G processor, the process of constructing records wrapper for a given search engine is done in 10 to 30 seconds. Once the wrapper is built, the record extraction from a new result page is done in a small fraction of a second.

In order to test the robustness of the generated wrapper, we compare the records extracted by our method with the test records acquired manually. The precision and recall measures are used to evaluate the result. 98% of all the records are extracted by program, with a precision of 99%. The precision indicates that the generated wrappers in our experiment are quite robust to acquire records. The recall is lower than the precision, which indicates that it sometimes misses a few records. The reason for this is that in the extraction step, the records different from more common ones are eliminated.

We compare our performance with the work in Zhao (2006), which addresses the issue of differentiating dynamic sections and records based on the sample result pages. It generates section wrappers by identifying section boundary markers in nine steps. It is more complicated in computation than ours because it renders each result page and extracts its content lines by a traversal of the DOM tree, while we use tag structure of a page. The accordance is making full use of the sample pages for given search engines. The method also gets a high precision of 98.8% and a recall of 98.7%.

7 Conclusion

The paper presents a novel method to acquire bilingual web pages automatically via search engines. In order to improve the efficiency and effectiveness, the snippets of search engines rather than the contents of the massive pages are analyzed to locate bilingual pages. Bilingual web page verification is modeled as a classification problem with word-overlap, length and frequency measures. Based on the similarity of HTML structures, AP clustering is used to extract web records from result pages of search engines. Experiments show that our algorithm has good performance in precision and recall.

As a valuable resource for up-to-date bilingual terms and sentences, bilingual web pages are counterpart to parallel monolingual web pages. Our method brings an efficient and effective solution to bilingual language engineering.

References

Adelberg B., NoDoSE. 1998. A tool for semi-Automatically extracting structured and sem-istructured data from text documents. In: *Proc.ACM SIGMOD Conference on man-agement of Data*, Seattle, WA (1998).

Baumgartner R., S. Flesca and G. Gottlob.2001. Visual Web Information Extraction with Lixto. *Proceedings of the 27th International Conference on Very Large Data Bases*, pp.119-128, September 11-14, 2001

Chang C., S. Lui. 2001. Information Extraction based on Pattern Discovery. In *Proceedings of the 10th international conference on World Wide Web*. pp.681-688, May 01-05, 2001, Hong Kong.

Chen Jiang and Jian-Yun Nie. 2000. Web

Parallel text mining for Chinese-English cross-language information retrieval. *Proceedings of RIAO2000 Content-Based Multimedia Information Access, CID, Paris*

Cortes, C. and V. Vapnik. 1995. Support-vector network. *Machine Learning 20*, pp.273-297.

Deng Dan. 2004. Research on Chinese-English word alignment. *Institute of Computing Technology Chinese Academy of Sciences*, Master Thesis. (in Chinese).

DuVerle David, Helmut Prendinger. 2009. A Novel Discourse Parser Based on Support Vector Machine Classification. *The 47th Annual Meeting of the Association for Computational Linguistics*. pp. 665-673

Frey B. J. and D. Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972-976.

Laender A, B. Ribeiro-Neto, A. da Silva, J. Teixeira. 2002. A Brief Survey of Web Data Extraction Tools. *ACM SIGMOD Record.* Volume 31, Number 2.

Liu B. and Y. Zhai. 2005. System for extracting Web data from flat and nested data records. In *Proceedings of the Conference on Web Information Systems Engineering*, pp.487-495.

Liu B., R. Grossman and Y. Zhai. 2003. Mining Data Records in Web Pages. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge Discovery and Data mining*, Washington, D.C, pp.601-606.

Liu Feifan, Jun Zhao, Bo Xu. 2003. Building Large-Scale Domain Independent Chinese-English Bilingual Corpus and the Researches on Sentence Alignment. *Joint Symposium on Computational Linguistics*.

Liu L., C. Pu and W. Han. 2000. An XML-Enabled Wrapper Construction System for Web Information Sources. *Proceedings of the 16th International Conference on Data Engineering*, pp.611.

Long Jiang, Shiquan Yang, Ming Zhou, Xiaohua Liu and Qingsheng Zhou. 2009. Mining Bilingual Data from the Web with Adaptively Learnt Patterns. *The 47th Annual Meeting of the Association for Computational Linguistics*. pp. 870-878 (2009)

Miao Gengxin, Junichi Tatemura, Wang-Pin Hsiung, Arsany Sawires, Louise E. Moser. 2009. Extracting data records from the web using tag path clustering. In *Proceedings of the 18th International Conference on World Wide Web*, Spain, Madrid.

Nie Jian-Yun, Michel Simard, Pierre Isabelle, Richard Durand 1999. Cross-Language Information Retrieval based on Parallel Texts and Automatic Mining of Parallel Texts in the Web. *SIGIR-1999*; 74-81.

Resnik Philip and Noah A. Smith. 2003. The web as a Parallel Corpus. *Computational Linguistics*.

Shi Lei, Cheng Niu, Ming Zhou, and Jianfeng Gao. 2006. A DOM Tree Alignment Model for Mining Parallel Data from the Web. In *Joint Proceedings of the Association for Computational Linguistics and the International Conference on Computational Linguistics*, Sydney, Australia.

White, R., Jose, J. & Ruthven, R. 2001.Query-biased web page summarisation: a task-oriented evaluation. In Proceedings *of the 24th ACM SIGIR Conference on Research and Development of Information Retrieval*. New Orleans, Louisiana, United States, pp. 412-413.

Zhai Y., B. Liu. 2005. Extracting Web Data Using Instance-Based Learning. *Web Information Systems Engineering*.

Zhai Y., B. Liu. 2005. Web Data Extraction Based on Partial Tree Alignment. In *Proceedings of the 14th international conference on World Wide Web.* May 10-14, 2005, Chiba, Japan.

Zhang Ying, Ke Wu, Jianfeng Gao, Phil Vines. 2006. Automatic Acquisition of Chinese-English Parallel Corpus from the web. In *Proceedings of 28th European Conference on Information Retrieval*.

Zhao H., W. Meng, Z. Wu, V. Raghavan, C. Yu. 2006. Automatic Extraction of Dynamic Record Sections From Search Engine Result Pages. In *Proceedings of the 32nd International conference on Very large databases*.