A DOM Tree Alignment Model for Mining Parallel Data from the Web

Lei Shi[1], Cheng Niu[1], Ming Zhou[1], and Jianfeng Gao[2]
[1]Microsoft Research Asia, 5F Sigma Center, 49 Zhichun Road, Beijing 10080, P. R. China
[2]Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
{leishi,chengniu,mingzhou,jfgao}@microsoft.com

Abstract

This paper presents a new web mining scheme for parallel data acquisition. Based on the Document Object Model (DOM), a web page is represented as a DOM tree. Then a DOM tree alignment model is proposed to identify the translationally equivalent texts and hyperlinks between two parallel DOM trees. By tracing the identified parallel hyperlinks, parallel web documents are recursively mined. Compared with previous mining schemes, the benchmarks show that this new mining scheme improves the mining coverage, reduces mining bandwidth, and enhances the quality of mined parallel sentences.

1 Introduction

Parallel bilingual corpora are critical resources for statistical machine translation (Brown 1993), and cross-lingual information retrieval (Nie 1999). Additionally, parallel corpora have been exploited for various monolingual natural language processing (NLP) tasks, such as word-sense disambiguation (Ng 2003) and paraphrase acquisition (Callison 2005).

However, large scale parallel corpora are not readily available for most language pairs. Even where resources are available, such as for English-French, the data are usually restricted to government documents (*e.g.,* the Hansard corpus, which consists of French-English translations of debates in the Canadian parliament) or newswire texts. The "governmentese" that characterizes these document collections cannot be used on its own to train data-driven machine translation systems for a range of domains and language pairs.

With a sharply increasing number of bilingual web sites, web mining for parallel data becomes a promising solution to this knowledge acquisition problem. In an effort to estimate the amount of bilingual data on the web, (Ma and Liberman 1999) surveyed web pages in the de (German web site) domain, showing that of 150,000 web-sites in the .de domain, 10% are German-English bilingual. Based on such observations, some web mining systems have been developed to automatically obtain parallel corpora from the web (Nie *et al* 1999; Ma and Liberman 1999; Chen, Chau and Yeh 2004; Resnik and Smith 2003; Zhang *et al* 2006). These systems mine parallel web documents within bilingual web sites, exploiting the fact that URLs of many parallel web pages are named with apparent patterns to facilitate website maintenance. Hence given a bilingual website, the mining systems use pre-defined URL patterns to discover candidate parallel documents within the site. Then content-based features will be used to verify the translational equivalence of the candidate pairs.

However, due to the diversity of web page styles and website maintenance mechanisms, bilingual websites use varied naming schemes for parallel documents. For example, the United Nation's website, which contains thousands of parallel pages, simply names the majority of its web pages with some computer generated ad-hoc URLs. Such a website then cannot be mined by the URL pattern-based mining scheme. To further improve the coverage of web mining, other patterns associated with translational parallelism are called for.

Besides, URL pattern-based mining may raise concerns on high bandwidth cost and slow download speed. Based on descriptions of (Nie *et al* 1999; Ma and Liberman 1999; Chen, Chau and Yeh 2004), the mining process requires a full host crawling to collect URLs before using URL patterns to discover the parallel documents. Since in many bilingual web sites, parallel documents are much sparser than comparable documents, a significant portion of internet bandwidth is wasted on downloading web pages without translational counterparts.

Furthermore, there is a lack of discussion on the quality of mined data. To support machine translation, parallel sentences should be extracted from the mined parallel documents. However, current sentence alignment models, (Brown *et al* 1991; Gale & Church 1991; Wu 1994; Chen

1993; Zhao and Vogel, 2002; *etc.*) are targeted on traditional textual documents. Due to the noisy nature of the web documents, parallel web pages may consist of non-translational content and many out-of-vocabulary words, both of which reduce sentence alignment accuracy. To improve sentence alignment performance on the web data, the similarity of the HTML tag structures between the parallel web documents should be leveraged properly in the sentence alignment model.

In order to improve the quality of mined data and increase the mining coverage and speed, this paper proposes a new web parallel data mining scheme. Given a pair of parallel web pages as seeds, the Document Object Model[1] (DOM) is used to represent the web pages as a pair of DOM trees. Then a stochastic DOM tree alignment model is used to align translationally equivalent content, including both textual chunks and hyperlinks, between the DOM tree pairs. The parallel hyperlinks discovered are regarded as anchors to new parallel data. This makes the mining scheme an iterative process.

The new mining scheme has three advantages: (i) Mining coverage is increased. *Parallel hyperlinks referring to parallel web page* is a general and reliable pattern for parallel data mining. Many bilingual websites not supporting URL pattern-based mining scheme support this new mining scheme. Our mining experiment shows that, using the new web mining scheme, the web mining throughput is increased by 32%; (ii) The quality of the mined data is improved. By leveraging the web pages' HTML structures, the sentence aligner supported by the DOM tree alignment model outperforms conventional ones by 7% in both precision and recall; (iii) The bandwidth cost is reduced by restricting web page downloads to the links that are very likely to be parallel.

The rest of the paper is organized as follows: In the next section, we introduce the related work. In Section 3, a new web parallel data mining scheme is presented. Three component technologies, the DOM tree alignment model, the sentence aligner, and the candidate parallel page verification model are presented in Section 4, 5, and 6. Section 7 presents experiments and benchmarks. The paper is finally concluded in Section 8.

[1] See http://www.w3.org/DOM/

2 Related Work

The parallel data available on the web have been an important knowledge source for machine translation. For example, *Hong Kong Laws*, an English-Chinese Parallel corpus released by Linguistic Data Consortium (LDC) is downloaded from the *Department of Justice of the Hong Kong Special Administrative Region* website.

Recently, web mining systems have been built to automatically acquire parallel data from the web. Exemplary systems include *PTMiner* (Nie et al 1999), STRAND (Resnik and Smith, 2003), BITS (Ma and Liberman, 1999), and PTI (Chen, Chau and Yeh, 2004). Given a bilingual website, these systems identify candidate parallel documents using pre-defined URL patterns. Then content-based features are employed for candidate verification. Particularly, HTML tag similarities have been exploited to verify parallelism between pages. But it is done by simplifying HTML tags as a string sequence instead of a hierarchical DOM tree. Tens of thousands parallel documents have been acquired with accuracy over 90%.

To support machine translation, parallel sentence pairs should be extracted from the parallel web documents. A number of techniques for aligning sentences in parallel corpora have been proposed. (Gale & Church 1991; Brown *et al.* 1991; Wu 1994) used sentence length as the basic feature for alignment. (Kay & Roscheisen 1993; and Chen 1993) used lexical information for sentence alignment. Models combining length and lexicon information were proposed in (Zhao and Vogel, 2002; Moore 2002). Signal processing techniques is also employed in sentence alignment by (Church 1993; Fung & McKeown 1994). Recently, much research attention has been paid to aligning sentences in comparable documents (Utiyama et al 2003, Munteanu et al 2004).

The DOM tree alignment model is the key technique of our mining approach. Although, to our knowledge, this is the first work discussing DOM tree alignments, there is substantial research focusing on syntactic tree alignment model for machine translation. For example, (Wu 1997; Alshawi, Bangalore, and Douglas, 2000; Yamada and Knight, 2001) have studied synchronous context free grammar. This formalism requires isomorphic syntax trees for the source sentence and its translation. (Shieber and Schabes 1990) presents a synchronous tree adjoining grammar (STAG) which is able to align two syn-

tactic trees at the linguistic minimal units. The synchronous tree substitution grammar (STSG) presented in (Hajic etc. 2004) is a simplified version of STAG which allows tree substitution operation, but prohibits the operation of tree adjunction.

3 A New Parallel Data Mining Scheme Supported by DOM Tree Alignment

Our new web parallel data mining scheme consists of the following steps:

(1) Given a web site, the root page and web pages directly linked from the root page are downloaded. Then for each of the downloaded web page, all of its anchor texts (*i.e.* the hyperlinked words on a web page) are compared with a list of predefined strings known to reflect translational equivalence among web pages (Nie *et al* 1999). Examples of such predefined trigger strings include: (i) trigger words for English translation {*English, English Version, 英文, 英文版, etc.*}; and (ii) trigger words for Chinese translation {*Chinese, Chinese Version, Simplified Chinese, Traditional Chinese, 中文, 中文版，, etc.*}. If both categories of trigger words are found, the web site is considered bilingual, and every web page pair are sent to Step 2 for parallelism verification.

(2) Given a pair of the plausible parallel web pages, a verification module is called to determine if the page pair is truly translationally equivalent.

(3) For each verified pair of parallel web pages, a DOM tree alignment model is called to extract parallel text chunks and hyperlinks.

(4) Sentence alignment is performed on each pair of the parallel text chunks, and the resulting parallel sentences are saved in an output file.

(5) For each pair of parallel hyperlinks, the corresponding pair of web pages is downloaded, and then goes to Step 2 for parallelism verification. If no more parallel hyperlinks are found, stop the mining process.

Our new mining scheme is iterative in nature. It fully exploits the information contained in the parallel data and effectively uses it to pinpoint the location holding more parallel data. This approach is based on our observation that parallel pages share similar structures holding parallel content, and parallel hyperlinks refer to new parallel pages.

By exploiting both the HTML tag similarity and the content-based translational equivalences, the DOM tree alignment model extracts parallel text chunks. Working on the parallel text chunks instead of the text of the whole web page, the sentence alignment accuracy can be improved by a large margin.

In the next three sections, three component techniques, the DOM tree alignment model, sentence alignment model, and candidate web page pair verification model are introduced.

4 DOM Tree Alignment Model

The Document Object Model (DOM) is an application programming interface for valid HTML documents. Using DOM, the logical structure of a HTML document is represented as a tree where each node belongs to some pre-defined node types (*e.g. Document, DocumentType, Element, Text, Comment, ProcessingInstruction etc.*). Among all these types of nodes, the nodes most relevant to our purpose are *Element* nodes (corresponding to the HTML tags) and *Text* nodes (corresponding to the texts). To simplify the description of the alignment model, minor modifications of the standard DOM tree are made: (i) Only the *Element* nodes and *Text* nodes are kept in our document tree model. (ii) The *ALT* attribute is represented as *Text* node in our document tree model. The *ALT* text are textual alternative when images cannot be displayed, hence is helpful to align images and hyperlinks. (iii) the *Text* node (which must be a leaf) and its parent *Element* node are combined into one node in order to concise the representation of the alignment model. The above three modifications are exemplified in Fig. 1.

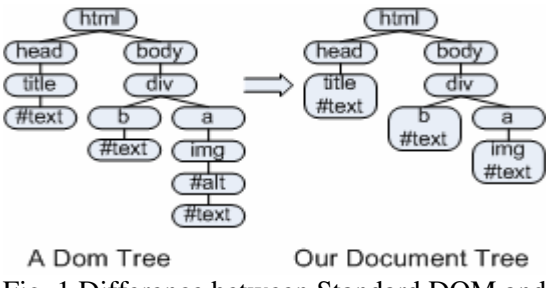

Fig. 1 Difference between Standard DOM and Our Document Tree

Despite these minor differences, our document tree is still referred as DOM tree throughout this paper.

4.1 DOM Tree Alignment

Similar to STSG, our DOM tree alignment model supports node deletion, insertion and substitution. Besides, both STSG and our DOM tree alignment model define the alignment as a tree hierarchical invariance process, *i.e.* if node *A* is aligned with node *B*, then the children of *A* are either deleted or aligned with the children of *B*.

But two major differences exist between STSG and our DOM tree alignment model: (i) Our DOM tree alignment model requires the alignment a sequential order invariant process, *i.e.* if node *A* is aligned with node *B*, then the sibling nodes following *A* have to be either deleted or aligned with the sibling nodes following *B*. (ii) (Hajic etc. 2004) presents STSG in the context of language generation, while we search for the best alignment on the condition that both trees are given.

To facilitate the presentation of the tree alignment model, the following symbols are introduced: given a HTML document D, $\mathrm{T^D}$ refers to the corresponding DOM tree; $\mathrm{N_i^D}$ refers to the $\mathrm{i^{th}}$ node of $\mathrm{T^D}$ (here the index of the node is in the breadth-first order), and $\mathrm{T_i^D}$ refers to the sub-tree rooted at $\mathrm{N_i^D}$, so $\mathrm{N_1^D}$ refers to the root of $\mathrm{T^D}$, and $\mathrm{T_1^D} = T^D$; $\mathrm{T_{[i,j]}^D}$ refers to the forest consisting of the sub-trees rooted at nodes from $\mathrm{T_i^D}$ to $\mathrm{T_j^D}$. $\mathrm{N_i^D}.t$ refers to the text of node $\mathrm{N_i^D}$; $\mathrm{N_i^D}.l$ refers to the HTML tag of the node $\mathrm{N_i^D}$; $\mathrm{N_i^D}.C_j$ refers to the $\mathrm{j^{th}}$ child of the node $\mathrm{N_i^D}$; $\mathrm{N_i^D}.C_{[m,n]}$ refers to the consecutive sequence of $\mathrm{N_i^D}$'s children nodes from $\mathrm{N_i^D}.C_m$ to $\mathrm{N_i^D}.C_n$; the sub-tree rooted at $\mathrm{N_i^D}.C_j$ is represented as $\mathrm{N_i^D}.TC_j$ and the forest rooted at $\mathrm{N_i^D}.C_{[m,n]}$ is represented as $\mathrm{N_i^D}.TC_{[m,n]}$. Finally *NULL* refers to the empty node introduced for node deletion.

To accommodate the hierarchical structure of the DOM tree, two different translation probabilities are defined:

$\Pr\left(T_m^F \middle| T_i^E\right)$: probability of translating sub-tree T_i^E into sub-tree T_m^F;

$\Pr\left(N_m^F \middle| N_i^E\right)$: probability of translating node N_i^E into N_m^F.

Besides, $\Pr\left(T_{[m,n]}^F \middle| T_{[i,j]}^E, A\right)$ represents the probability of translating the forest $T_{[i,j]}^E$ into $T_{[m,n]}^F$ based on the alignment *A*. The tree alignment *A* is defined as a mapping from target nodes onto source nodes or the null node.

Given two HTML documents F (in French) and *E* (in English), the tree alignment task is defined as searching for *A* which maximizes the following probability:

$$\Pr\left(A \middle| T^F, T^E\right) \propto \Pr\left(T^F \middle| T^E, A\right)\Pr\left(A \middle| T^E\right) \quad (1)$$

where $\Pr\left(A \middle| T^E\right)$ represents the prior knowledge of the alignment configurations.

By introducing p_d which refers to the probability of a source or target node deletion occurring in an alignment configuration, the alignment prior $\Pr\left(A \middle| T^E\right)$ is assumed as the following binominal distribution:

$$\Pr\left(A \middle| T^E\right) \propto (1-p_d)^L p_d^{\ M}$$

where *L* is the count of non-empty alignments in A, and *M* is the count of source and target node deletions in *A*.

As to $\Pr\left(T^F \middle| T^E, A\right)$, we can estimate as $\Pr\left(T^F \middle| T^E, A\right) = \Pr\left(T_1^F \middle| T_1^E, A\right)$, and $\Pr\left(T_l^F \middle| T_i^E, A\right)$ can be calculated recursively depending on the alignment configuration of A:

(1) If N_l^F is aligned with N_i^E, and the children of N_l^F are aligned with the children of N_i^E, then we have

$$\Pr\left(T_l^F \middle| T_i^E, A\right) = \Pr\left(N_l^F \middle| N_i^E\right)\Pr\left(N_l^F.TC_{[1,K]} \middle| N_i^E.TC_{[1,K']}, A\right)$$

where K and K' are degree of N_l^F and N_i^E.

(2) If N_l^F is deleted, and the children of N_l^F is aligned with T_i^E, then we have

$$\Pr\left(T_l^F \middle| T_i^E, A\right) = \Pr\left(N_l^F \middle| NULL\right)\Pr\left(N_l^F.TC_{[1,K]} \middle| T_i^E, A\right)$$

where K is the degree of N_l^F

(3) If N_i^E is deleted, and N_l^F is aligned with the children of N_i^E, then

$$\Pr\left(T_l^F \middle| T_i^E, A\right) = \Pr\left(T_l^F \middle| T_i^E.TC_{[1,K]}, A\right)$$

where K is the degree of N_i^E.

To complete the alignment model, $\Pr\left(T_{[m,n]}^F \middle| T_{[i,j]}^E, A\right)$ is to be estimated. As mentioned before, only the alignment configurations with unchanged node sequential order are considered as valid. So, $\Pr\left(T_{[m,n]}^F \middle| T_{[i,j]}^E, A\right)$ is estimated recursively according to the following five alignment configurations of A:

(4) If T_m^F is aligned with T_i^E, and $T_{[m+1,n]}^F$ is

aligned with $T^E_{[i+1,j]}$, then

$$\Pr\left(T^F_{[m,n]} \middle| T^E_{[i,j]}, A\right) = \Pr\left(N^F_m \middle| N^E_i\right) \Pr\left(T^F_{[m+1,n]} \middle| T^E_{[i+1,j]}, A\right)$$

(5) If T^F_m is deleted, and $T^F_{[m+1,n]}$ is aligned with $T^E_{[i,j]}$, then

$$\Pr\left(T^F_{[m,n]} \middle| T^E_{[i,j]}, A\right) = \Pr\left(N^F_m \middle| NULL\right) \Pr\left(T^F_{[m+1,n]} \middle| T^E_{[i,j]}, A\right)$$

(6) If T^E_i is deleted, and $T^F_{[m,n]}$ is aligned with $T^E_{[i+1,j]}$, then

$$\Pr\left(T^F_{[m,n]} \middle| T^E_{[i,j]}, A\right) = \Pr\left(T^F_{[m,n]} \middle| T^E_{[i+1,j]}, A\right)$$

(7) If N^F_m is deleted, and N^F_m's children $N^F_m.C_{[1,K]}$ is combined with $T^F_{[m+1,n]}$ to aligned with $T^E_{[i,j]}$, then

$$\Pr\left(T^F_{[m,n]} \middle| T^E_{[i,j]}, A\right) = \Pr\left(N^F_m \middle| NULL\right) \Pr\left(N^F_m.TC_{[1,K]} T^F_{[m+1,n]} \middle| T^E_{[i,j]}, A\right)$$

where K is the degree of N^F_m.

(8) N^E_i is deleted, and N^E_i's children $N^E_i.C_{[1,K]}$ is combined with $T^E_{[i+1,j]}$ to be aligned with $T^F_{[m,n]}$, then

$$\Pr\left(T^F_{[m,n]} \middle| T^E_{[i,j]}, A\right) = \Pr\left(T^F_{[m,n]} \middle| N^E_i.TC_{[1,K]} T^E_{[i+1,j]}, A\right)$$

where K is the degree of N^E_i.

Finally, the node translation probability is modeled as $\Pr\left(N^F_l \middle| N^E_j\right) \approx \Pr\left(N^F_l.l \middle| N^E_i.l\right) \Pr\left(N^F_l.t \middle| N^E_i.t\right)$. And the text translation probability $\Pr\left(t^F \middle| t^E\right)$ is model using IBM model I (Brown *et al* 1993).

4.2 Parameter Estimation Using Expectation-Maximization

Our tree alignment model involves three categories of parameters: the text translation probability $\Pr\left(t^F \middle| t^E\right)$, tag mapping probability $\Pr\left(l \middle| l'\right)$, and node deletion probability p_d.

Conventional parallel data released by LDC are used to train IBM model I for estimating the text translation probability $\Pr\left(t^F \middle| t^E\right)$.

One way to estimate $\Pr\left(l \middle| l'\right)$ and p_d is to manually align nodes between parallel DOM trees, and use them as training corpora for maximum likelihood estimation. However, this is a very time-consuming and error-prone procedure. In this paper, the inside outside algorithm presented in (Lari and Young, 1990) is extended to train parameters $\Pr\left(l \middle| l'\right)$ and p_d by optimally fitting the existing parallel DOM trees.

4.3 Dynamic Programming for Decoding

It is observed that if two trees are optimally aligned, the alignment of their sub-trees must be optimal as well. In the decoding process, dynamic programming techniques can be applied to find the optimal tree alignment using that of the sub-trees in a bottom up manner. The following is the pseudo-code of the decoding algorithm:

```
For i=|T^F| to 1 (bottom-up) {
    For j=|T^E| to 1 (bottom-up) {
        derive the best alignments among
        T_i^F.TC_[1,K_i] and T_j^E.TC_[1,K_j], and then
        compute the best alignment between
        N_i^F and N_j^E.
```

where $|T^F|$ and $|T^E|$ are number of nodes in T^F and T^E; K_i and K_j are the degrees of N^F_i and N^E_j. The time complexity of the decoding algorithm is $O(|T_F| \times |T_E| \times (\text{degree}(T^F) + \text{degree}(T^E))^2)$, where the degree of a tree is defined as the largest degree of its nodes.

5 Aligning Sentences Using Tree Alignment Model

To exploit the HTML structure similarities between parallel web documents, a cascaded approach is used in our sentence aligner implementation.

First, text chunks associated with DOM tree nodes are aligned using the DOM tree alignment model. Then for each pair of parallel text chunks, the sentence aligner described in (Zhao et al 2002), which combines IBM model I and the length model of (Gale & Church 1991) under a maximum likelihood criterion, is used to align parallel sentences.

6 Web Document Pair Verification Model

To verify whether a candidate web document pair is truly parallel, a binary maximum entropy based classifier is used.

Following (Nie *et al* 1999) and (Resnik and Smith, 2003), three features are used: (i) file length ratio; (ii) HTML tag similarity; (iii) sentence alignment score.

The HTML tag similarity feature is computed as follows: all of the HTML tags of a given web page are extracted, and concatenated as a string. Then, a minimum edit distance between the two tag strings associated with the candidate pair is computed, and the HMTL tag similarity score is defined as the ratio of match operation number to the total operation number.

The sentence alignment score is defined as the ratio of the number of aligned sentences and the total number of sentences in both files.

Using these three features, the maximum entropy model is trained on 1,000 pairs of web pages manually labeled as parallel or non-parallel. The Iterative Scaling algorithm (Pietra, Pietra and Lafferty 1995) is used for the training.

7 Experimental Results

The DOM tree alignment based mining system is used to acquire English-Chinese parallel data from the web. The mining procedure is initiated by acquiring Chinese website list.

We have downloaded about 300,000 URLs of Chinese websites from the web directories at *cn.yahoo.com, hk.yahoo.com and tw.yahoo.com.* And each website is sent to the mining system for English-Chinese parallel data acquisition. To ensure that the whole mining experiment to be finished in schedule, we stipulate that it takes at most 10 hours on mining each website. Totally 11,000 English-Chinese websites are discovered, from which 63,214 pairs of English-Chinese parallel web documents are mined. After sentence alignment, totally 1,069,423 pairs of English-Chinese parallel sentences are extracted.

In order to compare the system performance, 100 English-Chinese bilingual websites are also mined using the URL pattern based mining scheme. Following (Nie *et al* 1999; Ma and Liberman 1999; Chen, Chau and Yeh 2004), the URL pattern-based mining consists of three steps: (i) host crawling for URL collection; (ii) candidate pair identification by pre-defined URL pattern matching; (iii) candidate pair verification.

Based on these mining results, the quality of the mined data, the mining coverage and mining efficiency are measured.

First, we benchmarked the precision of the mined parallel documents. 3,000 pairs of English-Chinese candidate documents are randomly selected from the output of each mining system, and are reviewed by human annotators. The document level precision is shown in Table 1.

	URL pattern	DOM Tree Alignment
Precision	93.5%	97.2%

Table 1: Precision of Mined Parallel Documents

The document-level mining precision solely depends on the candidate document pair verification module. The verification modules of both mining systems use the same features, and the only difference is that in the new mining system the sentence alignment score is computed with DOM tree alignment support. So the 3.7% improvement in document-level precision indirectly confirms the enhancement of sentence alignment.

Secondly, the accuracy of sentence alignment model is benchmarked as follows: 150 English-Chinese parallel document pairs are randomly taken from our mining results. All parallel sentence pairs in these document pairs are manually annotated by two annotators with cross-validation. We have compared sentence alignment accuracy with and without DOM tree alignment support. In case of no tree alignment support, all the texts in the web pages are extracted and sent to sentence aligner for alignment. The benchmarks are shown in Table 2.

Alignment Method	**Number Right**	**Number Wrong**	**Number Missed**	**Precision**	**Recall**
Eng-Chi (no DOM tree)	2172	285	563	86.9%	79.4%
Eng-Chi (with DOM tree)	2369	156	366	93.4%	86.6%

Table 2: sentence alignment accuracy

Table 2 shows that with DOM tree alignment support, the sentence alignment accuracy is greatly improved by 7% in both precision and recall. We also observed that the recall is lower than precision. This is because web pages tend to contain many short sentences (one or two words only) whose alignment is hard to identify due to the lack of content information.

Although Table 2 benchmarks the accuracy of sentence aligner, but the quality of the final sentence pair outputs depend on many other modules as well, *e.g.* the document level parallelism verification, sentence breaker, Chinese word breaker, etc. To further measure the quality of the mined data, 2,000 sentence pairs are randomly picked from the final output, and are manually classified into three categories: (i) exact parallel, (ii) roughly parallel: two parallel sentences involving missing words or erroneous additions; (iii) not parallel. Two annotators are

assigned for this task with cross-validation. As is shown in Table 3, 93.5% of output sentence pairs are either exact or roughly parallel.

Corpus	Exact Parallel	Roughly Parallel	Not Parallel
Mined	1703	167	130

Table 3 Quality of Mined Parallel Sentences

As we know, the absolute value of mining system recall is hard to estimate because it is impractical to evaluate all the parallel data held by a bilingual website. Instead, we compare mining coverage and efficiency between the two systems. 100 English-Chinese bilingual website are mined by both of the system. And the mining efficiency comparison is reported in Table 4.

Mining System	Parallel Page Pairs found & verified	# of page downloads	# of downloads per pair
URL pattern-based Mining	4383	84942	19.38
DOM Tree Alignment-based Mining	5785	13074	2.26

Table 4. Mining Efficiency Comparison on 100 Bilingual Websites

Although it downloads less data, the DOM tree based mining scheme increases the parallel data acquisition throughput by 32%. Furthermore, the ratio of downloaded page count per parallel pair is 2.26, which means the bandwidth usage is almost optimal.

Another interesting topic is the complementarities between both mining systems. As reported in Table (5), 1797 pairs of parallel documents mined by the new scheme is not covered by the URL pattern-based scheme. So if both systems are used, the throughput can be further increased by 41%.

# of Parallel Page Pairs Mined by Both Systems	# of Parallel Page Pairs Mined by URL Patterns only	# of Parallel Page Pairs Mined by Tree Alignment only
3988	395	1797

Table 5. Mining Results Complementarities on 100 Bilingual Website

8 Discussion and Conclusion

Mining parallel data from web is a promising method to overcome the *knowledge bottleneck* faced by machine translation. To build a practical mining system, three research issues should be fully studied: (i) the quality of mined data, (ii) the mining coverage, and (iii) the mining speed. Exploiting DOM tree similarities helps in all the three issues.

Motivated by this observation, this paper presents a new web mining scheme for parallel data acquisition. A DOM tree alignment model is proposed to identify translationally equivalent text chunks and hyperlinks between two HTML documents. Parallel hyperlinks are used to pinpoint new parallel data, and make parallel data mining a recursive process. Parallel text chunks are fed into sentence aligner to extract parallel sentences.

Benchmarks show that sentence aligner supported by DOM tree alignment achieves performance enhancement by 7% in both precision and recall. Besides, the new mining scheme reduce the bandwidth cost by 8~9 times on average compared with the URL pattern-based mining scheme. In addition, the new mining scheme is more general and reliable, and is able to mine more data. Using the new mining scheme alone, the mining throughput is increased by 32%, and when combined with URL pattern-based scheme, the mining throughput is increased by 41%.

References

Alshawi, H., S. Bangalore, and S. Douglas. 2000. Learning Dependency Translation Models as Collections of Finite State Head Transducers. *Computational Linguistics*, 26(1).

Brown, P. F., J. C. Lai and R. L. Mercer. 1991. Aligning Sentences in Parallel Corpora. In *Proceedings of 29th Annual Meeting of the Association for Computational Linguistics.*

Brown, P. E., S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics,* V19(2).

Callison-Burch, C. and C. Bannard. 2005. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of 43th Annual Meeting of the Association for Computational Linguistics.*

Chen, J., R. Chau, and C.-H. Yeh. 1991. Discovering Parallel Text from the World Wide Web. In *Proceedings of the second workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalization.*

Chen, S. 1993. Aligning Sentences in Bilingual Corpora Using Lexical Information. In *Proceedings of 31st Annual Meeting of the Association for Computational Linguistics.*

Church, K. W. 1993. Char_align: A Program for Aligning Parallel Texts at the Character Level. In

Proceedings of 31st Annual Meeting of the Association for Computational Linguistics.

Fung, P. and K. Mckeown. 1994. Aligning Noisy Parallel Corpora across Language Groups: Word Pair Feature Matching by Dynamic Time Warping. In *Proceedings of the First Conference of the Association for Machine Translation in the Americas.*

Gale W. A. and K. Church. 1991. A Program for Aligning Sentences in Parallel Corpora. In *Proceedings of 29th Annual Meeting of the Association for Computational Linguistics.*

Hajic J., *et al.* 2004. Final Report: Natural Language Generation in the Context of Machine Translation.

Kay M. and M. Roscheisen. 1993. Text-Translation Alignment. *Computational Linguistics*, 19(1).

Lari K. and S. J. Young. 1990. The Estimation of Stochastic Context Free Grammars using the Inside-Outside Algorithm. *Computer Speech and Language,* 4:35—56, 1990.

Ma, X. and M. Liberman. 1999. Bits: A Method for Bilingual Text Search over the Web. In *Proceedings of Machine Translation Summit VII.*

Ng, H. T., B. Wang, and Y. S. Chan. 2003. Exploiting Parallel Texts for Word Sense Disambiguation: An Empirical Study. In *Proceedings of 41st Annual Meeting of the Association for Computational Linguistics.*

Nie, J. Y., M. S. P. Isabelle, and R. Durand. 1999. Cross-language Information Retrieval based on Parallel Texts and Automatic Mining of Parallel Texts from the Web. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development.*

Moore, R. C. 2002. Fast and Accurate Sentence Alignment of Bilingual Corpora. In *Proceedings of 5th Conference of the Association for Machine Translation in the Americas*.

Munteanu D. S, A. Fraser, and D. Marcu. D., 2002. Improved Machine Translation Performance via Parallel Sentence Extraction from Comparable Corpora. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004.*

Pietra, S. D., V. D. Pietra, and J. Lafferty. 1995. Inducing Features Of Random Fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

Resnik, P. and N. A. Smith. 2003. The Web as a Parallel Corpus. *Computational Linguistics*, 29(3)

Shieber, S. M. and Y. Schabes. 1990. Synchronous tree-adjoining grammars. In *Proceedings of the 13th International Conference on Computational linguistics.*

Utiyama, M. and H. Isahara 2003. Reliable Measures for Aligning Japanese-English News Articles and Sentences. In *Proceedings of 41st Annual Meeting of the Association for Computational Linguistics.*ACL 2003.

Wu, D. 1994. Aligning a parallel English-Chinese corpus statistically with lexical criterias. In *Proceedings of of 32nd Annual Meeting of the Association for Computational Linguistics.*

Wu, D. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3).

Yamada K. and K. Knight. 2001. A Syntax Based Statistical Translation Model. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics.*

Zhao B. and S. Vogel. 2002. Adaptive Parallel Sentences Mining From Web Bilingual News Collection. In *2002 IEEE International Conference on Data Mining.*

Zhang, Y., K. Wu, J. Gao, and Phil Vines. 2006. Automatic Acquisition of Chinese-English Parallel Corpus from the Web. In *Proceedings of 28th European Conference on Information Retrieval.*