Automatic Acquisition of Chinese–English Parallel Corpus from the Web

Ying Zhang[1], Ke Wu[2], Jianfeng Gao[3], and Phil Vines[1]

[1] RMIT University, GPO Box 2476V, Melbourne, Australia,
yzhang@cs.rmit.edu.au , phil@cs.rmit.edu.au
[2] Shanghai Jiaotong University, Shanghai 200030, China,
wuke@sjtu.edu.cn
[3] Microsoft Research, Redmond, Washington 98052, USA,
jfgao@microsoft.com

Abstract. Parallel corpora are a valuable resource for tasks such as cross-language information retrieval and data-driven natural language processing systems. Previously only small scale corpora have been available, thus restricting their practical use. This paper describes a system that overcomes this limitation by automatically collecting high quality parallel bilingual corpora from the web. Previous systems used a single principle feature for parallel web page verification, whereas we use multiple features to identify parallel texts via a k-nearest-neighbor classifier. Our system was evaluated using a data set containing 6500 Chinese–English candidate parallel pairs that have been manually annotated. Experiments show that the use of a k-nearest-neighbors classifier with multiple features achieves substantial improvements over the systems that use any one of these features. The system achieved a precision rate of 95% and a recall rate of 97%, and thus is a significant improvement over earlier work.

1 Introduction

Parallel corpora provide a rich source of translation information. In the past, they have been used to train statistical translation models [1–3], translation disambiguation systems [4], out-of-vocabulary term translation [5], and multilingual thesaurus construction [6]. However, some parallel corpora are subject to subscription or licence fee and thus not freely available, while others are domain specific. For example, parallel corpora provided by the Evaluations and Language resources Distribution Agency [7], the Linguistic Data Consortium [8], and the University Centre for Computer Corpus Research on Language [9], all require subscription or fee. There are several large manually constructed parallel corpora available on the web but they are always domain specific, thus significantly limiting their practical usage. For instance, the biblical text [10] in a number of languages (collected by the University of Maryland) and the European parliament proceedings parallel corpus (1996-2003) [11] in eleven European languages.

In order to make use of the ever increasing number of parallel corpora, a robust system is needed to automatically mine them from the web. This paper presents a system to automatically collect parallel Chinese–English corpora from the web — Web Parallel Data Extraction (WPDE). Similar to previous systems that have been developed for the same purposes, WPDE uses a three stage process: first, candidate sites are selected and crawled; second, candidate pairs of parallel texts are extracted; finally, we validate the parallel text pairs. Compared to previous systems, WPDE contains improvements at each stage. Specifically, in stage one, in addition to anchor text, image ALT text (the text that always provides a short description of the image and is displayed if an image is not shown) is used to improve the recall of candidate sites selection. In stage two, candidate pairs are generated by pattern matching and edit-distance similarity measure, whereas previous systems only applied one or the other of these. In stage three, where previous systems used a single principle feature to verify parallel pages, WPDE applies a KNN classifier to combine multiple features. Experiments on a large manually annotated data set show that each of the methods leads to improvements in terms of the overall performance in each step, and that the combined system yields the best overall result reported.

The structure of the paper is as follows. In Section 2, we consider other related work. Section 3 lays out the WPDE architecture. In Section 4 we detail our experiments and present the results we obtained; and Section 5 concludes the paper.

2 Related Work

The amount of information available on the web is expanding rapidly, and presents a valuable new source of parallel text. Recently, several systems have been developed to exploit this opportunity.

Nie et al. [1, 12] developed the PTMiner to mine large parallel corpora from the web. PTMiner used search engines to pinpoint the candidate sites that are likely to contain parallel pages, and then used the URLs collected as seeds to further crawl each web site for more URLs. The pairs of web pages were extracted on the basis of manually defined URL pattern-matching, and further filtered according to several criteria, such as file length, HTML structure, and language character set. Several hundred selected pairs were evaluated manually. Their results were quite promising, from a corpus of 250 MB of English–Chinese text, statistical evaluation showed that of the pairs identified, 90% were correct.

STRAND [13] is another well-known web parallel text mining system. Its goal is to identify pairs of web pages that are mutual translations. Resnik and Smith used the AltaVista search engine to search for multilingual websites and generated candidate pairs based on manually created substitution rules. The heart of STRAND is a structural filtering process that relies on analysis of the pages' underlying HTML to determine a set of pair-specific structural values, and then uses those values to filter the candidate pairs. Approximately 400

	Precision	Recall	Parallel text size	Number of pairs evaluated
PTMiner	90%	–	250 MB	100–200 (randomly picked)
STRAND	98%	61%	3500 pairs	400 (randomly picked)
PTI	93%	96%	427 pairs	all

Table 1. *Summarized Results from PTMiner, STRAND, and PTI*

pairs were evaluated by human annotators. STRAND produced fewer than 3500 English–Chinese pairs with a precision of 98% and a recall of 61%.

The Parallel Text Identification System (PTI) [14] was developed to facilitate the construction of parallel corpora by aligning pairs of parallel documents from a multilingual document collection. The system crawls the web to fetch (potentially parallel) candidate multilingual web documents using a web spider. To determine the parallelism between potential document pairs, a filename comparison module is used to check filename resemblance, and a content analysis module is used to measure the semantic similarity. The results showed that the PTI system achieves a precision rate of 93% and a recall rate of 96%. PTI is correct in 180 instances among a total of 193 pairs extracted. Our later evaluation showed that WPDE is able to produce 373 correct pairs with a precision of 97% and a recall of 94% on the same domain, using the file length feature-based verification only.

The summarized results from above studies are tabulated in Table 1.

3 The WPDE Architecture

WPDE is an automatic system for large scale mining of parallel text from existing English–Chinese bilingual web pages in a variety of domains. In summary, our procedure consists of three steps: candidate sites selection and crawling, candidate pairs extraction, and parallel pairs verification.

3.1 Candidate Sites Selection and Crawling

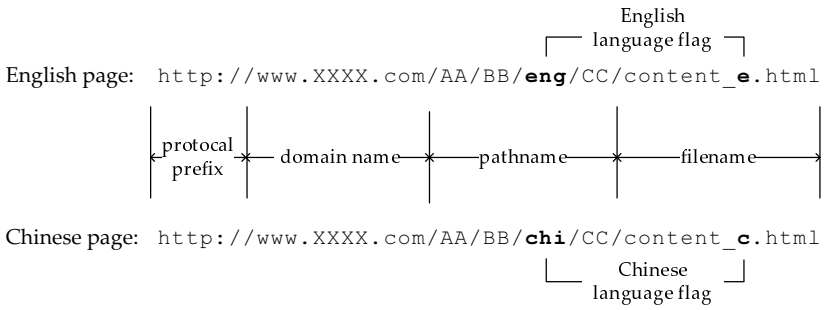
Rather than using search engines to identify the candidate sites, we started with a snapshot of two million web pages from Microsoft Research. We noticed that images representing the language types are almost always accompanied by their text equivalents — ALT text. One of the major differences between WPDE and previous systems is that the candidate sites are selected on the basis of both anchor text and image ALT text. For a given web page, we extract the hypertext links when the anchor text or the image ALT text matches a list of pre-defined strings that indicate English, simplified Chinese, and traditional Chinese (see Appendix A). If a website contains two or more hypertext links to the different versions, we select these as candidate websites. 1598 candidate

websites were selected based on the anchor text and 211 extra candidate websites were obtained using the image ALT text.

Once candidate sites were extracted from the snapshot, we used `Wget`[4] to fetch all documents from each site on the live web and create local copies of remote directory hierarchies.

3.2 Candidate Pairs Extraction

We then extract candidate parallel pairs from the crawled web pages. URLs consist of a protocol prefix, a domain name, a pathname, and a filename. Webmasters tend to name the pages with similar names if they are the translation of each other. The only difference between these two URLs is the segments that indicate the language type. For example, given the URLs of an English–Chinese parallel pair,



where `eng` and `e` are used to indicate the English version and `chi` and `c` are used to indicate the Chinese version. We observed that there are only five patterns `e,en,eng,engl,english` that are utilized to indicate the English version. Whereas, the patterns employed to indicate the Chinese version are quite unpredictable, and it is unrealistic to expect a "complete" pattern list. Therefore, previously employed language flag matching approaches [1, 12], that replace one language prefix/suffix/infix with all possible prefixes/suffixes/infixes in the other language based on a static pre-defined pattern list, will not work on a large scale URL matching process.

An improved approach combining pattern matching and edit-distance similarity measure [15] has been exploited in our work. For example, if an English pattern is detected in the pathname of an URL, we first extract the candidate Chinese URLs with the same protocol prefix, the same domain name, and the same pathname, except for the language flag segment. If the Chinese URL contains a language pathname segment that is in our standard Chinese pattern list — `c,ch,chi,chinese`, we select this URL. Otherwise we use an edit distance

[4] `http://www.gnu.org/software/wget/`

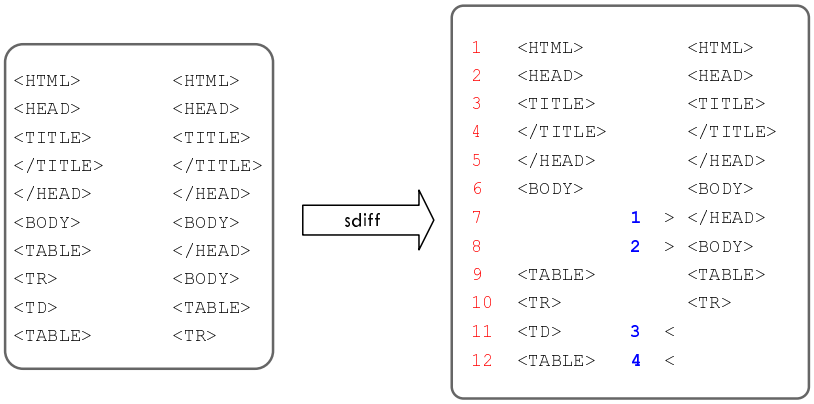


Fig. 1. *An example of file structure comparison using* `sdiff`.

metric to find the nearest match to one of these Chinese patterns, for example `tc,sc,tchi,schi`, etc. If the filenames are the same, the process is finished. Sometimes this is not the case, and an additional filename matching step is required. In the simplest case the filename will differ by one of the standard language flag patterns, otherwise we again use the same edit distance function to find the filename closest to the one of these Chinese patterns.

We have extracted a total of 7894 candidate pairs. Later evaluation showed that in isolation, this approach has a precision of 79%. Among a total of 606 pages, which are in *.pdf*, *.doc*, *.rtf*, and *.cfm* format, 558 of them are parallel pages with a high quality. We would suggest the web documents in these specific formats as a reliable parallel text source.

3.3 Parallel Pairs Verification

The candidate pairs extracted in the previous steps are further filtered based on three common features of parallel pages: the file length, the file structure, and the translation of the web page content. To filter out the pairs that are not similar enough, a threshold is set to each feature score. The experimental results are shown in Section 4.

File length. We assume the files sizes of Chinese–English parallel texts are roughly proportional. Additionally, files of length 40 bytes or less are discarded. Using these metrics, 323 candidate pairs (5%) were filtered out. For the candidate pairs that remain, we then calculate the ratio of the two file lengths $S_{len} =$ length(f_{ch}) / length(f_{en}). This ratio is then used in combination with other features as described below.

File structure. The HTML structures of two parallel pages should be similar. We extract the linear sequences of HTML tags from each candidate pair,

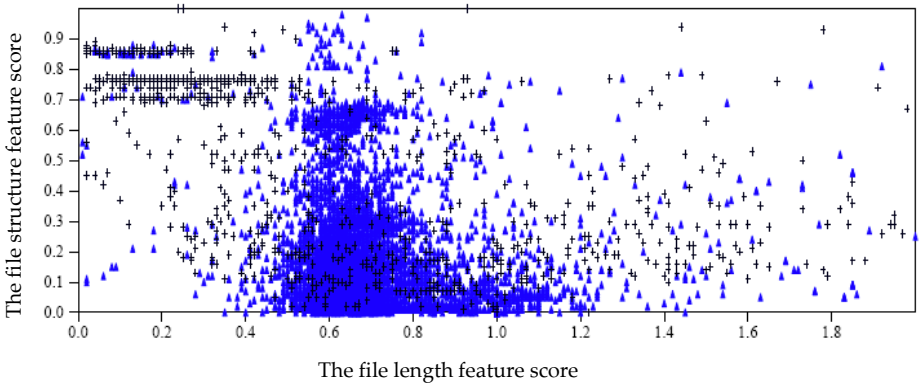


Fig. 2. *A scatter plot of the 2-feature dimensions. The x-axis shows the file length feature score. The y-axis shows the file structure feature score.*

then apply case-folding and remove noise, such as *meta*, *font* and *scripts*. Unix `sdiff`[5] is used to find differences between these two sequences of HTML tags obtained. For example, as shown in Figure 1, consider the two sequences of HTML tags on the left, the aligned sequence generated by `sdiff` is shown on the right.

The feature score of the file structure is calculated using $S_{struct} = N_{diff} / N_{all}$, where $N_{diff} = 4$ is the number of unaligned lines in the given example above, and $N_{all} = 12$ is the total number of the lines, and is used to normalize the score. Thus, the lower the score the better, with 0 being ideal.

Content translation. To consider the content translation of a candidate parallel pair, we align the two pages using the Champollion Tool Kit[6], which provides ready-to-use parallel text sentence alignment tools. Champollion depends heavily on lexical information, but uses sentence length information as well. Past experiments indicate that champollion's performance improves as the translation lexicon becomes larger. We therefore compiled a large English–Chinese lexicon, which contains 250,000 entries. The score of the content translation feature is calculated using $S_{trans} = N_{aligned} / N_{(ch,en)}$, where $N_{aligned}$ is the number of aligned sentences and $N_{(ch,en)}$ is the total number of lines in the two pages.

K-nearest-neighbors classifier. After investigating the recall-precision results of each single feature verification, we observed that although the file length feature produced the highest precision, the file structure feature can achieve a relatively high recall when lower precision is acceptable. Intuitively, it is possible to achieve better overall performance if multiple features can be combined using an appropriate model. To observe the data distribution in a 2-dimensional feature space, we generated the scatter plot matrix shown in Figure 2. The file

[5] `http://linuxcommand.org/man_pages/sdiff1.html`
[6] `http://champollion.sourceforge.net/`

length feature score is plotted in the X axis, while the file structure feature score is plotted on the Y axis. The 'true' pair is marked by triangle and the 'false' pair is represented by cross. As we can see, in the case of mixture of tightly clustered 'true and false' data, a linear decision boundary is unlikely to be optimal. k-nearest-neighbors method would be more appropriate for the mixture.

KNN has been successfully used for pattern classification on many applications [16]. Being a non-parametric classification method, it is a simple but effective method for classification. It labels an unknown sample with the label of the majority of the k nearest neighbors. A neighbor is deemed nearest if it has the smallest distance. The distance is usually calculated using the Euclidean distance.

Using a total of 6500 English-Chinese candidate pairs, we carried out tenfold cross-validation experiments using a KNN classifier to predict the correctness of a candidate pair. Specifically, the data is randomly split into 10 disjoint validation subsets, each with 650 pairs. In each fold, we then select one of those subsets as a test set with 650 test items and use the rest 5850 pairs as its training set; the fraction of true and false pairs in each fold's test and training sets approximates the overall division, 80% to 20%, respectively. The choice of k affects the performance of a KNN classifier. Wilson and Martinez [17] proposed that the k is typically a small integer that is odd and often determined from cross-validation. Therefore we choose the optimal k value with the best performance in cross-validation experiments. Through our experiments, we determined that the best results are generally obtained with $k = 15$ for 3-feature dimension, and $k = 7$ for 2-feature dimensions.

4 Experiments Results and Discussion

In this section, we describe the experimental setup and the experimental results.

4.1 Evaluation Methodology

The performance of a system that finds web parallel pages can be evaluated using standard IR measures of precision and recall. Precision represents the proportion of candidate parallel pages retrieved that are correct, thus:

$$Precision = \frac{Number\ of\ correctly\ aligned\ pairs}{Total\ number\ of\ aligned\ pairs}$$

Whereas recall represents the proportion of parallel pages that the system actually found:

$$Recall = \frac{Number\ of\ correctly\ aligned\ pairs}{Total\ number\ of\ parallel\ pairs\ in\ the\ collection}$$

Recall can be calculated for a test collection since the total number of parallel pairs can be determined by inspection, but cannot be calculated for the entire web.

RUN ID	Precision	Recall
RUN_{len} ($0.55 \leq \text{S}_{len} < 0.75$)	97%	70%
RUN_{struct} ($\text{S}_{struct} \leq 0.1$)	95%	46%
RUN_{trans} ($\text{S}_{trans} \geq 0.1$)	90%	53%

Table 2. *Effect of the features separately. For the file length feature, ratios between 0.55 and 0.75 achieved the best precision. For the file structure feature, pairs with scores* ≤ 0.1 *performed best, whereas for the translation feature, attribute scores* ≥ 0.1 *provided the best precision.*

We used three Chinese–English bilingual speakers (none of whom are authors of this paper) to evaluate the correctness of all the parallel pairs we extracted from the web. Only if the English and Chinese pages contain entirely the same meaning, the pair is annotated as a 'correct pair'. While previous systems have been evaluated on relatively small data set (about a few hundreds of pairs), we created a large manually annotated test collection containing around 6500 English–Chinese pairs.

4.2 Web Crawling Results

A total of 61 web sites, which include 26 *.hk* sites and 35 *.cn* sites, were randomly selected from the candidate websites obtained in Section 3.1. We have crawled about 2.7 GB of web data, comprising approximately 53,000 web pages. We noticed that the quality of the parallel data provided by the *.hk* sites is seemingly better than that provided by the *.cn* sites, and therefore we strongly suggest that more importance should be attached to the *.hk* web sites in candidate website selection.

4.3 Parallel Pairs Mining Results

We then tested the effect of the features, both separately and in various of combinations.

Single feature effect. We have run three experiments to separately gauge the effectiveness of each of these features — the file length, the file structure, and the content translation features in RUN_{len}, RUN_{struct}, and RUN_{trans}, respectively. The evaluation results with the highest average precision achieved using tenfold cross-validation are shown in Table 2.

Surprisingly, the file length feature, the simplest and thus the most efficient, is clearly superior. When $0.55 \leq \text{S}_{len} < 0.75$, we are able to achieve a precision of 97% and a recall of 70%. This compares favorably to the results of STRAND and PTMiner (see Table 1), which while not directly comparable because of the the differing corpora, suggests that our system performs reasonably well.

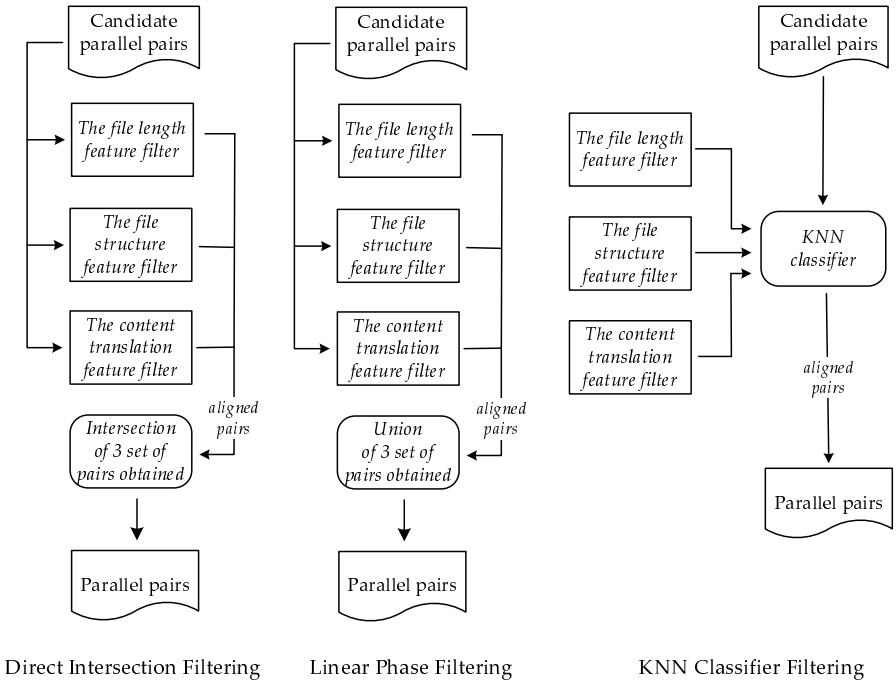


Fig. 3. *Outline of different feature fusion methods.*

Our utilization of linear sequence of HTML tags to determine whether two pages are parallel, is similar to that of STRAND and PTMiner. The file HTML structure feature provides a relatively high precision; meanwhile, it greatly impairs the recall.

The content translation feature has produced mediocre results. Given Champollion depends heavily on lexical information (previously described in Section 3.3), we suspect the main reason is that the majority of the candidate pairs we have generated in Section 3.2 are in traditional Chinese, where the bilingual lexicon we have compiled is based on simplified Chinese. Although there are no differences between the basic vocabularies or grammatical structures of simplified and traditional Chinese, different Chinese communities translate English terms in different ways. Due to the limited communication between mainland China (using simplified Chinese) and Taiwan, Hong Kong and the overseas areas (using traditional Chinese), there are some differences in terminology, especially new cultural or technological nouns. For instance, the English computer phrase "cross-language information retrieval" is commonly translated in simplified Chinese as "跨语言信息检索", while in traditional Chinese it is "跨語言資訊檢索". This suggests that better results might be obtained if specially tailored lexicons were used for mainland and overseas Chinese text.

RUN ID	Features			Precision	Recall
	F_{len}	F_{struct}	F_{trans}		
RUN_{len} (Baseline)	√			97	70
RUN_{inters}	√	√		97	30
	√		√	97	64
		√	√	97	27
	√	√	√	98	20
RUN_{linear}	√	√		95	85
	√		√	95	88
		√	√	94	89
	√	√	√	96	90
RUN_{knn}	√	√		94	94
	√		√	94	97
		√	√	93	97
	√	√	√	95	97

Table 3. *Effect of the different types of feature fusion. (All values are percentages.)*

Feature fusion effect. This set of experiments allowed us to test whether using feature fusion in the parallel pairs verification is likely to provide any benefit, as well as the effect of the number of the features of fusion on the overall performance. As shown in Figure 3, three types of feature combinations are investigated: the direct intersection, the linear phase filtering, and a KNN classifier.

In the direct intersection run RUN_{inters}, we evaluated a direct intersection of the pair sets aligned by each of the features. In the linear phase filtering run RUN_{linear}, the candidate pairs were passed through the linear phase filters. The pairs that are unable to be detected by the first feature filter were aligned using the second feature filter, the pairs left were piped to the last feature filter and processed. In other words, this process produces the union of the sets of pairs aligned by each filter. In the RUN_{knn}, we experimented with a KNN classifier previously described in Section 3.3. For example, using a feature space of three dimensions each pair instance x is represented as a vector $\langle S_{len}(x), S_{struct}(x), S_{trans}(x) \rangle$. RUN_{len} provided the best results for a single feature run, and thus is used to establish a reference by which we can measure our feature fusion results. The results reported are obtained after selecting an optimal threshold for each of the feature scores. The experimental results with the highest average precision achieved using tenfold cross-validation are shown in Table 3.

The results of the direct intersection combination method (RUN_{inters}) were disastrous. This suggests a large proportion of correct pairs only satisfy some of the above three features. The result of this was often that many correct pairs were omitted. This outcome is corroborated by the results of RUN_{linear}. Using

the liner phase filtering feature fusion, we are able to achieve a precision of 96% and a recall of 90%. The KNN classifier further improved the recall to 97%. We used the Wilcoxon ranked signed test to test the statistical significance of the improvement. It showed a significant improvement at the 95% confidence level, and emphasizes the importance of a good feature fusion technique.

Our experiments also show that 3-feature fusion statistically significantly outperforms 2-feature fusion in both RUN_{linear} and RUN_{knn}. Therefore we conclude that a larger number of features will increase the overall performance of the system.

5 Conclusion

The paper describes WPED, an automatic mining system for bilingual web parallel corpora. This system used several new techniques to extract parallel web pages, and thus has the potential to find more candidate pages than previous systems. We have explored the use of multiple features via a KNN classifier. Experimental results show that the use of the KNN classifier with multiple features achieves substantial improvements over the systems that use any one of these features. WPDE has achieved a precision rate of 95% and a recall rate of 97%, and thus is a significant improvement over earlier work.

6 Acknowledgments

This work was done while the first and second authors were visiting Microsoft Research Asia. We thank Professor Jian-Yun Nie for his valuable discussion and advice.

References

1. Nie, J.Y., Simard, M., Isabelle, P., Durand, R.: 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 in Information Retrieval, Berkeley, California, United States, ACM Press (1999) 74–81
2. Franz, M., McCarley, J.S., Ward, T., Zhu, W.J.: Quantifying the utility of parallel corpora. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, United States, ACM Press (2001) 398–399
3. Brown, P.F., Cocke, J., Pietra, S.D., Pietra, V.J.D., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. Computational Linguistics **16** (1990) 79–85
4. Ballesteros, L., Croft, W.B.: Resolving ambiguity for cross-language retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, ACM Press (1998) 64–71

5. McEwan, C.J.A., Ounis, I., Ruthven, I.: Building bilingual dictionaries from parallel web documents. In: Proceedings of the 24th BCS-IRSG European Colloquium on IR Research, London, UK, Springer-Verlag (2002) 303–323
6. Chau, R., Yeh, C.H.: Construction of a fuzzy multilingual thesaurus and its application to cross-lingual text retrieval. In: Proceedings of the 1st Asia-Pacific Conference on Web Intelligence: Research and Development, Maebashi City, Japan, Springer-Verlag (2001) 340–345
7. `http://www.elda.org/`.
8. `http://www.ldc.upenn.edu/`.
9. `http://www.comp.lancs.ac.uk/computing/research/ucrel/`.
10. `http://www.umiacs.umd.edu/users/resnik/parallel/bible.html`.
11. `http://people.csail.mit.edu/koehn/publications/europarl/`.
12. Kraaij, W., Nie, J.Y., Simard, M.: Embedding web-based statistical translation models in cross-language information retrieval. Computational Linguistics **29** (2003) 381–419
13. Resnik, P., Smith, N.A.: The web as a parallel corpus. Computational Linguistics **29** (2003) 349–380
14. Chen, J., Chau, R., Yeh, C.H.: Discovering parallel text from the world wide web. In: Proceedings of the 2nd Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation, Dunedin, New Zealand, Australian Computer Society, Inc. (2004) 157–161
15. Lowrance, R., Wagner, R.A.: An extension of the string-to-string correction problem. Journal of the ACM **22** (1975) 177–183
16. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **13** (1967) 21–27
17. Wilson, D.R., Martinez, T.R.: Instance pruning techniques. In: Proceedings of the 14th International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1997) 403–411

A A List of Pre-defined Strings

`english`
`chinese`
`simplifiedchinese`
`chinesesimplified`
`traditionalchinese`
`chinesetraditional`
`englishversion`
`simplifiedchineseversion`
`traditionalchineseversion`

英文	英文首页
简体	中文首页
繁體	中文简体
英文版	中文繁體
中文版	简体中文
简体版	简体中文版
繁體版	繁體中文
英文网站	繁體中文版
中文网站	