Crawling Chinese-Myanmar Parallel Corpus: Automatic Collection, Screening and Cleaning Corpus

Kai Xiong¹, Rui Yuan¹, Wenxue He¹,∗, Yanmei Jing¹, Yansheng Wang¹, Qiqi He² and Huafu Li¹

¹School of Physics and Electronic Information, Yunnan Normal University, Kunming, 650500, China
²School of Information Science and Technology, Yunnan Normal University, Kunming, 650500, China

∗E-mail: wendell31132@hotmail.com

Abstract. The collection of Chinese-Myanmar Parallel Corpus (CMPC) is the key step in the natural language processing (NLP) and training Machine Translation Engine (MTE) of Southeast Asia minority languages. As the scarcity of CMPC resources that efficient corpus collection methods are worth studying extremely. Traditional corpus collection methods include manual collection, text recognition of books and Internet crawlers, etc. Among them, the most efficient method to collect corpus is internet crawler preached by many. Traditional Internet crawler algorithm is interfere easily by a lot of spamming and advertising that lead to the time-consuming and low-precision. We propose a web crawler mechanism combines acquisition automatically technology bilingual website list, crawling corpus and cleaning corpus to obtain high quality parallel corpus. Firstly, using the hyperlinks to recursively access related corpus websites through building the website graph. Furthermore, the breadth-first, Backline and PageRank crawler framework used to build a corpus selection model based on crawling with threshold, matching link, ranking the heat of page, through this, the CMPC can be found accurately. Finally, the corpus cleaning model based on the HTML parsing to determine a set of standardized token sequences. By testing the Chinese-Myanmar reptile algorithm established in this paper, the experimental results show that our benchmarks this model exceeds previous published benchmarks. Up to now, we have obtained 1.1 million parallel corpus pairs of Chinese-Myanmar.

1. Introduction
The CMPC is a one-to-one correspondence between the Burmese and its corresponding Chinese translation. It provides indispensable training data for training the Chinese-Myanmar (CM) machine translation model [1], and is also an important basic resource for lexicography [2] and cross-language information retrieval [3-5].

The two traditional methods obtained by the Chinese-Myanmar candidate website are: Method 1: Using a search engine to collect the target URL. PTMiner [6], STRAND [7] and can obtain a large number of links through search engines and anchor texts, and then use these links as seed links which to recursive downloads. WPDE [8] adds ALT information for the picture. Such methods can automatically discover candidate sites, but the candidate resources are mixed and contain a large amount of irrelevant page site information. Method 2 defines the target source. BITS [9] pre-collects a number of related topics, and then recursively downloads. Although this method can effectively find
the list of candidate websites, the number of selected web pages is limited and the selection of the website needs manual intervention, and the website cannot be automatically and quickly determined. This paper proposes a website heuristic collection method, which solves the problem that the source resources of CMPC are small and the coverage of the field is limited. The initial URL seed set used to collect CMPC crawlers must be highly relevant to the topic defined in advance. It does not have to collect all the web pages, only crawl those pages related to the theme [10-11], generally three ways to develop a crawling strategy for web pages. 1. Depth-first traversal [12], the method's crawl logic can lead to excessive depth, so that the crawled pages lack corresponding value, and will also affect the crawl hit rate and crawl efficiency. 2. Best-first traversal [13], this method does not necessarily crawl the optimal selection every time, and may even bypass the optimal page. 3. Breadth-first traversal [14], the pages crawled by this method are closer to the links in the seed set. Pages are more likely to be required and less likely to be repeated. However, the efficiency of the algorithm is reduced after downloading and filtering a large number of pages. Therefore, this paper proposes its own crawling algorithm based on breadth-first traversal combined with backlink $I_d(p)$ and PageRank $I_d(p)$.

This paper proposes a program to collect CMPC from the web, including the discovery and acquisition of CMCW, web crawlers based on Java framework, collection and cleaning corpus. The main advantages of the program in this paper (i)Previous studies mostly obtained corpus resources from a specific bilingual website (government website or conference website), and the collected resources are often small in scale and limited in scope. We first confirms the CM website list and uses it as a heuristic information to use the engine to obtain valuable bilingual hybrid websites, thus expanding the source of bilingual parallel corpus. (ii)This paper develops a crawling priority system based on the Java framework, completes the collection of CMCW from the Web, and downloads the candidate website pages. Using the threshold capture and stop operation mode, the crawler program preferentially accesses the high I(p) page, avoiding the overlap of the download pages, thereby reducing the browsing and downloading of irrelevant pages, and the crawling efficiency is improved. (iii)After obtaining the CMCW, use the regular expression processing mechanism to remove the labels in the original corpus text, standardize the text format, symbols, and obtain high-quality parallel statements. The CMPC automatic acquisition system is not only easy to construct and maintain, but also able to acquire data accurately and quickly.

2. Chinese-Myanmar parallel website acquisition
In this chapter, we present a new solution to obtain the candidate website of CMPC. First and foremost, we represent the CMPC website as the graph structure. Furthermore, we propose a site acquisition model to identify cross-text information and hyperlinks between two parallel crawl processes. By crawling the identified parallel hyperlinks, we can recursively obtain a list of CMPC websites and download candidate website pages. At the same time, we optimized the traditional parallel crawling process, increased the download rate, and avoided the overlap of corpus pages.

Due to the complexity of web page style and website maintenance mechanism, the CM website uses different naming schemes for parallel documents, which also makes it difficult to perform crawling based on URL patterns. Therefore, we propose a new parallel corpus acquisition mechanism. The mechanism use the CM parallel website list as seed information to represent each website list information as Crawl-proc. Then, using the graph alignment structure to communicate and coordinate between the website lists, which discover new parallel hyperlinks and enrich the website list of the CMPC. When downloading pages within multiple retrieval processes, the same page may be downloaded repeatedly. To coordinate the process and prevent overlap, we optimized the acquisition engine. In figure 1, we show the framework for the Chinese-Myanmar parallel website.
Figure 1. CMPC website acquisition framework

The graph alignment structure completes the communication and coordination between the website lists. Using the graph model, the logical structure of our corpus page is represented as a graph where each node belongs to a predefined node type. In all nodes, we assume that the Web has S₁ and S₂ partitions. At the same time, each Crawl-proc crawl from the root page a and f of each site. When Crawl-proc processes the inter-partition link, C₁ notifies page g after downloading page a (and c), and C₂ transfers the URL of page d to C₁ after downloading page h. The alignment model is shown in figure 2.

3. Crawling Chinese-Myanmar parallel corpus

In this section, we use the hypertext information from the corpus page we obtained in the previous section. A model for crawling CMPC models and corpus cleaning models was proposed. First of all, the traditional method of crawling CMPC is to use automatic language tagging. This method removes all interfaces that are not CM language, resulting in the loss and confusion of a large number of parallel corpora. In order to solve this problem, we added a CM corpus selection model based on threshold crawling, link matching, and page heat ranking. We used many sorting structure directories of web sites to extract the most relevant CM language resources. Secondly, one of us corpus cleaning model is proposed, which relies on the parsing of the underlying HTML of the page to determine a set of standardized token sequences. The first step in this process is to linearize the HTML document. Parallel corpora generates a linear sequence of three tokens through a common set of labels; the second step is to align the linearized sequence using standard dynamic programming techniques, including normalization, tokenization stemming, and stemming and lemmatization techniques.

3.1. Chinese-Myanmar corpus selection model

Due to the lack of small language resources, search engines cannot retrieve every web interface. Therefore, it is very important to develop effective crawling strategies to prioritize CM corpus pages.

First, we designed the CM corpus selection model. Among them, we put forward the page importance metrics, and judge the relevance of the page to the corpus, page priority and corpus standards. At the same time, we use threshold crawling and detection to detect page hits and visits. Page importance metrics:

- \( I(p) \): Number of pages.
- \( I_s(p, Q) \): A measure of text similarity between web pages p and Q.
- Inverse document frequency: Inverse document frequency can be multiplied by the number of times the word appears in the document to calculate the page importance.
- Number of backlinks \( I_b(p) \): Number of web URLs that the search engine points to the page.
- G-Threshold condition, any page with \( I(p) \geq G \) is called a "hot" page.
Since the system sorts the pages after the crawl is complete, we calculate the PageRank as the page rank using a term weighting strategy that calculates the inverse document frequency for all crawl pages.

\[ I_r(p) = (1-d) + d \sum_{i=1}^{n} \frac{IR_i}{c_i} \]  

(1)

where \( t_{[0..n]} \) is the set of pages running to time \( t \), \( c_i \) is the number of backlinks of page \( t_i \), and \( d \) is the probability of random access to the next page, referred to as the damping factor.

In order to keep the crawler’s page vector space near the corpus list, we added an average similarity rating between the page and the list.

\[ \text{similarity} = \frac{\sum_{p \in C(i)} \sqrt{\left( \sum_{k: p \cap w} w_{pk} \right) \left( \sum_{k: v \cap w} w_{vk} \right)}}{|C(i)|} \]  

(2)

where \( V \) is the vector representing the list, \( w_{pk} \) is the TF-IDF weight of term \( k \) in \( p \) where inverse document frequency is computed from the crawl set \( C(i) \) once vital pages have been visited by each crawler.

As each crawler page set grows over time, we can map the reptile's trajectory over time and evaluate the system. In order to make our experiments easy to manage, we consider that the CM corpus page constitutes the entire Web, and only evaluate the performance in this case. Figure 3 shows a simplified control flow chart for the CM corpus selection model, which consists mainly of a parallel page selection engine and a crawler.

Second, we implement and evaluate the corpus selection model framework by combining three types of crawling-first, Backlink and PageRank crawling algorithms. Figure 4 shows our web crawler algorithm. Among them, the crawler manages three main data structures:

- The URL queue contains URLs that have been viewed and need to be accessed. Once the page is accessed, it is stored (including its URL) in the crawled page face. \( I_r(p, Q) \) - A measure of text similarity between web pages \( p \) and \( Q \).
- Links contain form pairs (\( u1, u2 \)), where URL \( u2 \) is displayed as URL \( u1 \) in the visited page.
- Number of backlinks \( I_b(p) \) - Number of web URLs that the search engine points to the page.

3.2. Chinese-Myanmar corpus cleaning model

![Figure 3. Simplified control flow chart of Chinese-Myanmar corpus selection model](image_url)

![Figure 4. Web crawler algorithm](image_url)
After crawling the CMPC, we obtained the entire HTML source code. This is what the browser needs to render the page. But most of this is useless to us. We need to use regular expressions to extract all the plain text that is visible on the crawled page surface. Simultaneously, "page extraction" and "post-processing, persistence" are the two stages of the crawler. Separating the modules can make the code structure clearer, which can be executed on separate threads and on different machines. Therefore, we define a CM corpus cleaning model responsible for the processing of the extracted results, including calculations, persistence to files, and databases. It relies on parsing the underlying HTML of the page to determine a set of standardized token sequences. In the actual CM parallel website, we found that many authors use tags to format text, which would destabilize the corpus for nonlinear structures. Therefore, we use the fast linear structure alignment algorithm to complete the parsing of HTML. Standard dynamic programming techniques align linearized sequences, including normalization, tokenization, stemming, and lemmatization techniques. The CM corpus cleaning model extracts titles and sentences from the corpus interface and displays them. Then we save them all in the database.

4. Experimental result

4.1. Web list and Targets Dataset
In order to evaluate our candidate website list acquisition algorithm, we started running random breadth-first crawling through sites such as Burma library to collect candidate website lists and target data sets. Since our dataset was downloaded by the crawler in a specific way, it may not be able to correctly represent the actual situation. Table 1 shows a few sample topics.

There are the precision is used to evaluate the web list performance. The formulations are as follows:

$$\text{Precision} = \frac{\text{AP}}{\text{TP}}$$

where AP denotes the number of appropriate pages, TP denotes the total number of pages, $0 \leq \text{Precision} \leq 1$$

| weblist             | Description                          | Targets                                      |
|---------------------|--------------------------------------|----------------------------------------------|
| Daily newspapers    | Kyemon (The Mirror) newspaper        | http://www.kicnews.org/                      |
|                     | Myanma Alin (The Light of Myanmar)   | http://burma.irrawaddy.org/                  |
|                     | New Light of Myanmar                 | http://burmese.dvb.no/                       |
|                     | The Yadanabon                        | http://www.bbc.co.uk/burmese/                |
| Digital media list  | Duwun                                 | http://www.duwun.com.mm/                     |
|                     | Newday Myanmar                       | http://www.newdaymyanmar.com/                |
| Published overseas  | Mizzima News Agency                  | http://mizzimaburmese.com/                   |
|                     | Mandalay Gazette                     | http://www.mandalaygazette.com/              |
|                     | Freedom News Group                   | http://www.freedomnewsgroup.com/             |

In the next experiment, we used the crawler program designed in this article to download these lists: we tested the download efficiency of the crawler program by using 20 pages from the 2, 8, 32, and 64 process downloads in figure 5. After that, we analyze the score of hot pages crawling at a given point, the $P_{ST}$, We consider using the threshold condition G for crawling in 3, 10 and 100 in Section 3 According to the popular page definition, $H$ is 806,200 (58%), 291,900 (21%) and 140,390 (10.1%) pages were considered popular among the 1,390,000 pages crawled. In figure 6, the
horizontal axis is part of the Burmese webpage that crawls over time. At the right end of the horizontal axis, all 1,390,000 pages are accessed.

The experimental results show that as our definition of popular pages becomes more restrictive (larger $G$), the crawler can find popular pages faster. Even if $G$ is large, it is always difficult to find the "last set" of popular pages. That is, on the right side of 0.8 on the horizontal axis, the reptile finds the popular interface at roughly the same rate as the random reptile.

Then we looked at all the documents to see if they were suitable for the corpus and completed the evaluation of the list of networks. The result is shown in table 2.

**Table 2.** Web list precision results

| web list          | processes | Total       | Appropriate | Total       | Appropriate |
|-------------------|-----------|-------------|-------------|-------------|-------------|
|                   |           | Docs | Words | Docs | % | Words | % |
| Daily newspapers  | 2         | 57   | 2,321,888 | 47   | 82.46% | 1,189,459 | 51.23% |
|                   | 8         | 64   | 2,521,212 | 42   | 65.63% | 1,079,394 | 42.81% |
|                   | 32        | 65   | 2,853,710 | 29   | 44.62% | 921,968    | 32.31% |
|                   | 64        | 67   | 2,887,755 | 25   | 37.31% | 827,535    | 28.66% |
|                   | 2         | 77   | 2,934,642 | 67   | 87.01% | 1,570,224 | 53.51% |
| Digital media list| 8         | 82   | 2,987,674 | 57   | 69.51% | 1,337,166 | 44.76% |
|                   | 32        | 85   | 3,022,155 | 41   | 48.24% | 1,031,087 | 34.12% |
|                   | 64        | 85   | 3,089,788 | 37   | 43.53% | 981,462    | 31.76% |
|                   | 2         | 28   | 1,606,312 | 22   | 78.57% | 791,682    | 49.29% |
| Published overseas| 8         | 31   | 1,723,220 | 18   | 58.06% | 672,612    | 39.03% |
|                   | 32        | 33   | 1,801,123 | 15   | 45.45% | 589,459    | 32.73% |
|                   | 64        | 35   | 1,832,001 | 13   | 37.14% | 523,428    | 28.57% |

**Figure 5.** The precision that use web list approach.

**Figure 6.** Fraction of web crawled.

### 4.2. Web crawl experiment

In this section, we evaluate the performance and efficiency of the crawling algorithm described in Section 2. There are two general-purpose experimental evaluation approaches to exhibits the performances of our approaches. The first metric is the simple recall level based on the target pages:

$$
recall = \frac{|Ci \cap T|}{|T|}
$$

(4)
where $C(i)$ is the set of pages crawled by crawler $i$ and $T$ is the target set. This measure allows us to determine how well a crawler can locate a few highly relevant pages. However, there may well be many other relevant pages that are not specified in the target set. To assess a crawler’s capability to locate these other relevant pages, we have to estimate the relevance of any crawled page. We do so using the lexical similarity between a page and the list to guide link selection. The second metric we measure is the mean similarity between the list and the set of pages crawled. It has been defined in section 3. Both recall and similarity can be plotted against the number of pages crawled to obtain a trajectory over time that displays the behavior of the crawl. We now outline the results of a number of crawls carried out to evaluate and compare performance and efficiency for the crawling algorithms described in Section 3.

The performance of the three crawlers – PageRank, Breadth-First, Backlink, is compared in figure 7 and figure 8. We ran three variations of each crawling algorithm for values of our model. Recall and similarity yield qualitatively consistent results. As expected Breadth-First performs poorly and thus constitutes a baseline. It is no surprise that PageRank performs optimum performance.

5. Contribution

Aiming at the practical problems that CM parallel sentences are difficult to obtain at low cost and high efficiency, we propose a crawler framework that automatically obtains a list of bilingual corpus websites, corpus crawling and corpus cleaning. The framework mainly includes: 1 the engine and website list automatically obtains the CM corpus candidate website, 2 combines the breadth-first, Backlink and PageRank crawler algorithms to implement the CM corpus crawling, 3 parses the HTML to determine a set of standardized token sequences. Compared with other traditional methods, this method has the advantage of automatically obtaining a large number of effective CM websites and broadening the source of corpus. Compared with other traditional methods, this method has the advantage of automatically obtaining a large number of effective CM websites and broadening the source of corpus. The web crawler uses the regular expression processing mechanism to filter irrelevant pages quickly and efficiently capture high-quality CMPC. The conclusion is that the method proposed in this paper is indeed effective in the acquisition of CMPC with scarce information. The method can extract parallel sentence pairs on the CM parallel website with low cost and high efficiency, and provide the basis for Chinese and Myanmar natural language processing and machine translation engine training. In future research, we need to improve the adaptability and crawling efficiency of the framework, optimize the intelligent capture and proofreading capabilities of the method to quickly build a large-scale parallel corpus.

Acknowledgment

This work was supported in part by the research is funded by the National Natural Science Foundation of China 61841113.
References

[1] Brown P F, Cocke J, Pietra S, et al. A statistical approach to machine translation[J]. Computational Linguistics, 1990, 16(2):79-85.

[2] Sun Le,Jin Y B, Du Lin S, et al. Automatic Extraction of Bilingual Term Lexicon from Parallel Corpora[J]. Journal Of Chinese Information Processing, 2000, 14(6): 33-39.

[3] Zhao xin, Deng Ming-xing Web-based Cross Language Information Retrieval Technique[J], Computer and Modernization, 2009, (2): 35-38.

[4] Chen J, Nie J Y. Automatic construction of parallel English-Chinese corpus for cross-language information retrieval[C]// Applied Natural Language Processing Conference. 2000:21-28.

[5] Lu Y, Zhang X, Zheng D. Automatic English-Chinese Parallel Corpus Acquisition and Sentences Extraction[J]. Journal of Computational Information Systems, 2013, 9(6):2365-2372.

[6] Nie J Y, Simard M, Isabelle P, et al. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the Web[J]. 1999:253-269.

[7] Resnik, P.and N. A. Smith. The Web as a Parallel Corpus[J]. Computational Linguistics, 2003, 29(3):223-248.

[8] 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 Internationalization, Dunedin, New Zealand, Australian Computer Society, Inc. 2004:157-161

[9] Yi S, Shi S, Wang H, et al. Parallel Approach and Platform for Large-Scale WEB Data Extraction[C]// International Conference on Advanced Cloud & Big Data. 2013.

[10] Furche T, Gottlob G, Grasso G, et al. OXPath: A language for scalable data extraction, automation, and crawling on the deep web[J]. VLDB Journal, 2013, 22(1):47-72.

[11] Baroni M, Bernardini S. BootCaT: Bootstrapping Corpora and Terms from the Web[C]//LREC. 2004: 1313-1316.

[12] Bazin A. A depth-first search algorithm for computing pseudo-closed sets[J]. Discrete Applied Mathematics, 2018.

[13] Kishimoto A, Fukunaga A, Botea A. Parallel Best-First Search for Optimal Sequential Planning (Research Statement)[J]. 2009.

[14] Feng Z, Lin H, Zhai J, et al. An adaptive breadth-first search algorithm on integrated architectures[J]. Journal of Supercomputing, 2018(3–4):1-21.