Journal of Education and Educational Research
ISSN: 2957-9465 | Vol. 16, No. 2, 2025

A Methodological Framework for Collecting and Processing Yi Monolingual and Yi-Chinese Bilingual Corpora

Junping Huang *

School of Chinese Language and Literature, Southwest Minzu University., Chengdu Sichuan, 610041, China
*** Corresponding author Email:** hip2049@126.com

Abstract: Addressing the challenges of collecting and processing Yi monolingual and Yi-Chinese bilingual corpora, this study proposes leveraging Optical Character Recognition and web crawler technologies to collect corpus from printed publications and the World Wide Web. The process encompasses corpus denoising, proofreading, bilingual sentences alignment, deduplication, and word segmentation, culminating in corpus storage. The theoretical framework presented herein offers valuable insights for the collection and processing of corpus, while the constructed corpora can serve as a resource for research in Yi character information processing, Yi-Chinese translation, and Yi-Chinese bilingual language ontology.

Keywords: Yi Character; Yi-Chinese Bilingualism; Corpus Collection; Corpus Processing.

1. Introduction

Since the State Council approved the implementation of the "The Standardization Scheme for Yi character" in 1980, the standardized Yi character (hereinafter referred to as Yi character) has been effectively promoted in the northern dialect region of the Yi character. Over more than four decades of development, a rich array of corpus resources has emerged, encompassing online content, newspapers, and books, including some Yi-Chinese bilingual corpora (Yi character and its corresponding Chinese translations). With the continuous growth of Yi character information and the rapid advancement of information technology, the influence of AI-related technologies on Yi character research has been increasingly significant. Regarding research on Optical Character Recognition (OCR) for the Yi character, Haipeng S et al. [1] proposed deep learning-based method for detection and recognition of Yi characters. Shang W et al.[2] centered on Yi printed character recognition using a deep learning model. Wang Chengping[3][4] conducted exploration and research on the word segmentation of Yi language and the construction of the Yi language corpus. As a low-resource language, Yi faces a scarcity of corpora, which has become one of the major bottlenecks for researchers studying Yi or Yi-Chinese bilingualism. This study will employ OCR and web crawler technologies to collect corpus from printed publications and the World Wide Web, and will conduct noise removal, proofreading, bilingual sentence alignment, deduplication, Word Segmentation, and storage of the corpus. On the one hand, this approach will broaden the channels for collecting and processing Yi monolingual and Yi-Chinese bilingual corpora; On the other hand, it will facilitate the research, education, preservation, and transmission of the Yi language.

2. Process Analysis

First, select the paper publications and Yi character websites. Next, train an OCR model for the Yi character using Yi character corpus. Then, collect the corpus from the website and publications through web crawling and OCR. After that, perform corpus denoising, proofreading, bilingual sentence alignment, deduplication, and word segmentation. Finally, store the processed corpus. The specific workflow is illustrated in Figure 1.

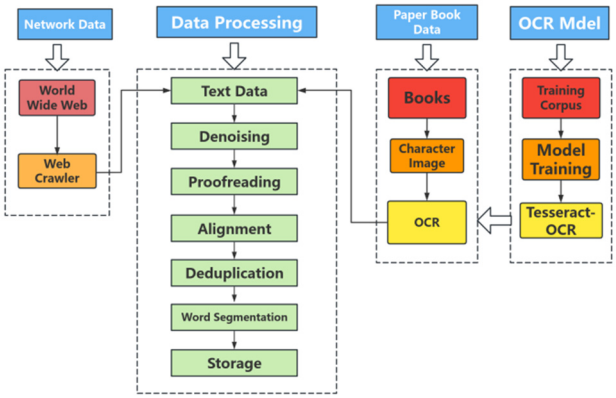


Figure 1. Flow chart of corpus collection and processing

3. Corpus Collection

The corpus is collected using the following two methods: Firstly, the corpus is obtained from paper publications through OCR technology. Secondly, it is crawled from the World Wide Web using web crawler technology.

3.1. OCR

3.1.1. Yi character OCR

Tesseract-OCR is chosen to train a Yi character OCR model, and this model is employed to collect Yi character corpus. Tesseract-OCR is an open-source OCR library, renowned for its high accuracy and flexibility, enabling it to be trained to recognize specific types of script. PyTesseract, an OCR recognition library, encapsulates Tesseract-OCR with a Python API. Hence, prior to using PyTesseract, it is necessary to install Tesseract-OCR and set up the appropriate running environment. Tesseract-OCR V5.3.0, a deep learning recognition engine leveraging the Long Short-Term Memory

(LSTM), significantly enhances the accuracy of character recognition.

Four commonly used Yi fonts, namely Baiti, *Songti*, Heiti and Xiheiti, are selected as the research objects. A total of 1,520,196 Yi characters, numbers, letters, and punctuation marks were collected from Yi language books and newspapers covering news, literature, science and technology, agriculture, medicine, and education, which includes 1,440,646 Yi characters. Tesseract-OCR[5] provides detailed instructions: Initially, a Starter-Trained-Data needs to be generated. Then, character images are created from the training dataset containing 1,520,196 characters. Lastly, complete the training of the Yi character OCR model.

To verify the effectiveness of the OCR model. First, prepare 10 character images of the same content for each of the four fonts: Baiti, *Songti*, Heiti and Xiheiti. Then, perform OCR on these images, obtaining 3,631, 3,612, 3,625, and 3,627 Yi characters from the four fonts respectively. Through analysis and statistics of the recognition results, determine that the recognition accuracy rate is the percentage of correctly recognized Yi characters out of the total number of Yi characters. The number of incorrectly or missed recognized Yi characters is included in the total number of characters that could not be correctly recognized. The recognition accuracy rates for the Baiti and *Songti* are between 93% and 94%, while those for the Heiti and Xiheiti are slightly higher, at over 97%. The OCR program was written in Python, utilizing the Pytesseract library to call the Yi character OCR module, thereby completing the OCR and output the Yi character text.

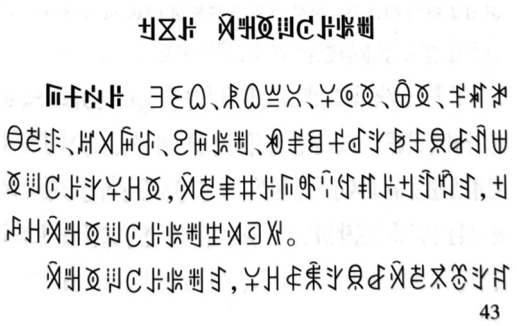


Figure 2. Image of some Yi characters in "ꆏꀋꌠꋍꂷꆏ"

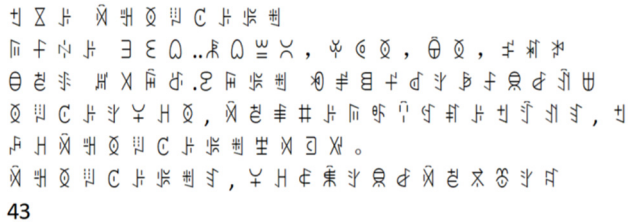


Figure 3. Text collected by Yi OCR

Figure 2 presents the image of the Yi characters to be recognized, while Figure 3 displays the corpus obtained after the OCR of Yi characters. Upon comparing Figures 2 and 3, it is observed that the Yi character "ꑌ" in the third from the end position of the fourth line in Figure 3 is incorrectly identified as "ꑍ". Additionally, some punctuation marks are also misidentified, such as semicolons being recognized as dots. After completing the OCR, the recognition results for each character image are separately stored in a distinct file.

3.1.2. Chinese OCR

The acquisition of Chinese corpus utilizes relatively mature and easily accessible tools, specifically exploring Tesseract-OCR, WeChat, and WPS Office for OCR of Chinese character images. The characteristics of these three software programs are as follows: (1) Tesseract-OCR has a slightly lower recognition accuracy compared to the latter two software; (2) While WeChat boasts high recognition accuracy, it can only process a character image at a time; (3) WPS Office exhibits high accuracy in OCR and is capable of recognizing multiple character images simultaneously. All three methods of character image recognition satisfy the research requirements. However, to enhance the efficiency of OCR, WPS Office was chosen for the Chinese OCR of character images. Similar to the storage method of the Yi character corpus, the recognition results for each character image are separately stored in individual files.

3.2. Web Crawler

3.2.1. Web Crawler Concept

A web crawler crawls across the web, at its core is an element of recursion, and it retrieves page contents for a URL, examines that page for another URL, and retrieves that page [6]. The depth-first search strategy is employed for web crawling. This strategy involves using a specified web page address as the entry point, analyzing all web addresses contained within that page, and adding any newly discovered web addresses to the list of web addresses. If a web address already exists in the list, it will not be added. The system selects unvisited web addresses from the list for access until all addresses on the list have been visited and processed. To prevent the crawler from getting stuck due to depth-first search, if a web page contains no Yi character content, the web addresses within that page will not be added to the list of web addresses.

3.2.2. Corpus Crawling

To ensure the accuracy and standardization of the crawled corpus, the Yi People's Network (Yi version) and People's Network (Yi version) are selected as the sources for corpus collection, encompassing news, literature, art, and other Yi monolingual corpus. Python's third-party libraries, Requests and Beautiful Soup, are employed to implement the web crawler. The Requests library is easy to use and features short execution times, significantly reducing program runtime and enabling browser simulation for access to web addresses. The web crawler employs the GET method to retrieve web page data, utilizes the requests.get function to obtain specific web page data by sending a request to the designated web address. The Beautiful Soup library serves as a data parsing tool for web pages, capable of extracting necessary data from web pages. The web crawler's steps are as follows:

(1) Import third-party libraries: The Requests library reads web page data, the Re library utilizes Regular Expressions to access specific web page data, the Time library is used for file naming, and the BeautifulSoup library extracts data from HTML or XML files.

(2) The main function controls the process and invokes other functions. It inputs the initial target web page address and sequentially extracts other web page addresses contained within.

(3) The designated web address is accessed and data is crawled by simulating a Firefox browser. If the crawling data is successful, the crawled data is returned; otherwise, "Crawling Failed" is output.

(4) Access the web address and retrieve the web data, selecting Yi monolingual and Yi-Chinese bilingual data

through regular expressions.

(5) Call the function to create a file and save the data crawled from the web page. The file name is generated based on the creation time. First, create a file in the specified path, then store the crawled data in the file, and finally close the file.

(6) Determine the tags of web pages containing web addresses and filter out those without web addresses. Finally, add all the web addresses under the web pages to the web address list 1. If the web address cannot be accessed successfully or the web content does not contain Yi characters, and the web address already exists in the web address list, the web address will not be added to address list 1.

(7) Traverse all web page addresses in the web address list 1, and save the data of a web page if it meets the set conditions. The web address list 2 stores the web addresses that have been visited in the address list 1. Each time a web address in the address list 1 is visited, the web address is added to address list 2, deleted from address list 1, and the iteration continues until all the web addresses in the address list 1 are processed. Repeat steps 3, 4, and 5 for each page read. Crawled content is mainly Yi monolingual corpus, with supplementary Yi-Chinese parallel corpus.

4. Corpus Processing

4.1. Corpus Preprocessing

(1) Corpus cleaning: The corpus collected by the web crawler contains some invalid information, which is deleted by designing a program and utilizing Regular Expressions to remove redundant information.

(2) Proofreading: In the corpus collected via OCR, a small portion of characters and punctuation marks may be incorrectly recognized or missed, necessitating manual proofreading, correction, and supplementation.

(3) Sentence alignment: Yi sentences and their corresponding Chinese translations were saved in an Excel table in a manual way to achieve alignment.

(4) Corpus deduplication: In the Excel spreadsheet, select the column containing the Yi sentences. Under the "Home" tab, click "Conditional Formatting" > "Highlight Cells Rules" > "Duplicate Values". This operation will highlight the duplicates in the table. Finally, manually delete the highlighted rows to remove duplicates.

4.2. Word Segmentation

4.2.1. Yi Word Segmentation

For dictionary-based Yi character segmentation, it is first necessary to construct a Yi character segmentation dictionary. This study obtains Yi vocabulary from relevant Yi dictionaries and collected neologisms, removes duplicates, and ends up with a Yi segmentation dictionary containing 65,551 words.

The Maximum Matching algorithm is a dictionary-based word segmentation method that relies on a pre-constructed dictionary. Qu Y H et al.[7] proposed a revised BMM and RMM algorithm for Chinese automatic word segmentation. This algorithm is easy to operate and requires no training, only preparing the dictionary in advance is necessary. This algorithm encompasses Forward Maximum Matching (FMM), Reverse Maximum Matching (RMM), and Bi-directional Maximum Matching (BMM). The FMM algorithm segments sentences from left to right. For instance, consider the sample sentence "ꑌꑍꑟꑱꑷꑸ". Assuming the maximum word length predefined in the dictionary is 4 characters, the first candidate word identified during initial matching is assumed to span 4 characters. The sentence is scanned from left to right, yielding the substring "ꑌꑍꑟꑱ". As this substring does not exist in the dictionary, the matching process fails at this stage. Consequently, the final character "ꑱ" is truncated, and matching proceeds iteratively. This operation is repeated until the substring "ꑌꑍ" is successfully segmented. Subsequently, scanning and segmentation proceed in the same manner until the remaining string length reaches zero, at which point the process terminates. The final segmentation result is "ꑌꑍ/ꑟ/ꑱꑷꑸ/ꑹ". The principle of the RMM algorithm is analogous to that of the FMM algorithm, with the key distinction being its right-to-left segmentation direction. During the matching process, words are matched from the right end of the sentence. For example, when segmenting the sample sentence "ꑌꑍꑟꑱꑷꑸꑹ" (with the maximum word length predefined as 4 characters in the dictionary), the first candidate word identified via left-to-right maximum matching was initially assumed to span 4 characters. However, upon scanning from right to left, the resulting substring "ꑱꑷꑸꑹ" did not correspond to any entry in the dictionary. The leftmost character "ꑱ" is then truncated and this operation is repeated iteratively until the single-character substring "ꑹ" is successfully segmented. Subsequently, scanning and segmentation proceed in the same manner until the remaining string length reaches zero. The segmentation result is "ꑌꑍ/ꑟ/ꑱꑷꑸ/ꑹ". The BMM algorithm involves running both the FMM and RMM algorithms separately to obtain the segmentation results of each. The results of these two algorithms are then compared: If the number of word segments differs, the result with fewer segments is adopted; If they are the same, either result can be chosen.

This study utilizes Python to design a Yi character word segmentation program, and employs the Yi character word segmentation dictionary and the BMM algorithm to segment the Yi character corpus. The word segmentation effect for the Yi character corpus is illustrated in Figure 4.

```
D:\python\python3.10\python.exe D:\py-program\segmentation.py
input:ꀀꀋꀔꀘꀞꀢꀤꀦꀪꀬꀮꀰꀲꀴꀶꀸꀺꀼꀾꁀꁂꁄꁆꁈꁊꁌꁎ
Forward maximum macthing word segmentation
['ꀀꀋꀔ', 'ꀘꀞ', 'ꀢꀤ', 'ꀦꀪ', 'ꀬꀮ', 'ꀰ', 'ꀲ', 'ꀴ', 'ꀶꀸ', 'ꀺꀼ', 'ꀾꁀ', 'ꁂ', 'ꁄꁆ', 'ꁈ', 'ꁊꁌꁎ']
Backward maximum macthing word segmentation
['ꁎ', 'ꁆꁈ', 'ꁄ', 'ꀾꁂ', 'ꀾ', 'ꀺꀼ', 'ꀀꀋ', 'ꀴ', 'ꀲ', 'ꀰ', 'ꀬꀮ', 'ꀢꀤ', 'ꀘꀞ', 'ꀀꀋꀔ']
Best segmentation is:
ꀀꀋꀔ/ꀘꀞ/ꀢꀤ/ꀦꀪ/ꀬꀮ/ꀰ/ꀲ/ꀴ/ꀶꀸ/ꀺꀼ/ꀾꁀ/ꁂ/ꁄꁆ/ꁈ/ꁊꁌꁎ/
```

Figure 4. The effect of Yi word segmentation

4.2.2. Chinese Word Segmentation

Jieba[8] is an excellent Chinese word segmentation component for Python. It's easy to install, delivers accurate segmentation, and is both open-source and free. Additionally,

users can customize the segmentation by adding a custom dictionary. Jieba offers four segmentation modes: Precise, Full, Search Engine, and Paddle. This study considered the pros and cons of these modes and the specific data requirements of the research, chose Jieba's Precise Mode for word segmentation, and utilized the Jieba word segmentation module in Python to segment the Chinese corpus. The segmentation effect is shown in Figure 5.

```
import jieba
str = "自古彝族人民就有赛马的风俗。"
seg_list = jieba.cut(str,cut_all=False)
print("Segmentation outcome: " + "/ ".join(seg_list))
Segmentation outcome:自古/彝族/人民/就有/赛马/的/风俗。
```

Figure 5. The effect of Chinese word segmentation

4.3. Corpus Storage

As an open-source relational database management system, MySQL has been widely adopted for its stable performance and ease of use [9]. Therefore, MySQL is chosen to store and manage the corpus. To facilitate the retrieval and application of the corpus, it is necessary to classify the corpus, which can be roughly divided into the following two categories and five subcategories: the two categories are Yi monolingual data and Yi-Chinese bilingual data. Under each category, there are five subcategories: news, culture, literature, science, and others. In the Yi monolingual data, each group of data for each subcategory includes Yi sentences and their corresponding segmented sentences. In the Yi-Chinese bilingual data, each group of data in each subcategory includes Yi sentences, segmented Yi sentences, Chinese sentences, and segmented Chinese sentences.

For the design of the MySQL database, two separate databases were established to store Yi monolingual data and Yi-Chinese bilingual data. Within each database, five tables were created, named News, Culture, Literature, Science, and Others. Each table contained five fields: Order, Yi, Yi Segmentation, Chinese, and Chinese Segmentation. The Order field uniquely stores the sequence number for each set of data. The Yi field holds the Yi corpus, while the Yi Segmentation field stores the segmented Yi corpus. Similarly, the Chinese field stores the Chinese corpus, and the Chinese Segmentation field holds the segmented Chinese corpus. Once the database was constructed, the corpus was imported, completing the storage process. Statistics show that the Yi monolingual database contains 13,538 Yi sentences, while the Yi-Chinese bilingual database has 11,982 sentence pairs.

5. Conclusion and Prospect

Given the challenges in collecting and processing Yi monolingual and Yi-Chinese bilingual corpora, this study utilizes OCR and a web crawler to collect the data, preprocess the corpus through cleaning, proofreading, bilingual sentence alignment, and deduplication. The two-way maximum matching algorithm and Jieba word segmentation are employed to segment the Yi and Chinese sentences respectively. Ultimately, MySQL facilitates the storage of the corpus. The theoretical framework of this study offers technical insights for the collection and processing of Yi monolingual and Yi-Chinese bilingual corpora. The established corpus serves as a valuable resource for training and testing in areas such as word segmentation, Part-of-Speech Tagging, Named Entity Recognition, sentiment analysis, Information Retrieval, Yi-Chinese Machine Translation, and Large Language Models in Yi Natural Language Processing. Additionally, it provides corpus support for Yi-Chinese translation studies and the research of the Yi-Chinese bilingual language ontology. Future efforts could focus on image processing technologies to enhance the performance of Yi OCR. In addition, there remains significant research potential in Yi word segmentation, Yi-Chinese Bilingual Alignment, and the storage and management of corpora.

References

[1] Haipeng S, Xueyan D, Jian S, et al. A Method for Detecting and Recogni-zing Yi Character Based on Deep Learning [J]. Computers, Materials & Con-tinua,2024,78(2):2721-2739. DOI:10.32604/CMC.2024.046449.

[2] Shang W, Chen L, Xiang W. Yi printed character recognition based on de-ep learning[J].Procedia Computer Science, 2024, 242584-591.DOI:10.1016/J.PROCS.2024.08.111.

[3] Cheng Ping W. The Research on Method and Difficulty Analysis of Autom-atic Word Segmentation in Yi Language for Information Processing[J]. The Frontiers of Society, Science and Technology,2019,1(8).

[4] Chengping W. The Research and Construction of Yi Corpus for Informatio-n Processing[J]. International Journal of New Developments in Engineering and Society,2019,3(4).

[5] https://github.com/tesseract-ocr/.

[6] Mitchell, R. Web Scraping with Python, 2nd ed.; No Starch Press: Sebasto-pol, CA, USA, 2018; pp. 33–48.

[7] Qu Y H, Zhao W. A Revised BMM and RMM Algorithm of Chinese Aut-omatic Words Segmentation[J].Advanced Materials Research,2011,1286(267-267):199-204.DOI:10. 4028/ www. scientific.net/AMR.267.199.

[8] https://github.com/fxsjy/jieba.

[9] Zhou Q, Zhang J. Research on the Design and Implementation of Case T-eaching in MySQL Database Course[J].Education Reform and Development,2025,7(5):145-150.DOI:10. 26689/ ERD. V7I5.10588.