Research on Methods of Parsing and Classification of Internet Super Large-scale Texts

Mengting Song, Hang Zheng, Zhen Tao, Jia Jiang, Bin Pan*

Hunan University of Finance and Economics, Changsha 410205, China

*676506201@qq.com

Abstract: Web crawlers are an important part of modern search engines. With the development of the times, data has shown explosive growth, and mankind has entered a "big data era". For example, Wikipedia, which carries knowledge achievements from all over the world, records real-time news that occurs every day and provides users with a good text search database[1]. Wikipedia updates data up to 50+GB every day. This project focuses on solving the problems of data acquisition and data analysis. At the same time, it downloads and parses the latest data of Wikipedia and analyzes XML files, and then uses SVM algorithm and Naive Bayes algorithm to classify articles, Train the model to download Wikipedia files efficiently and parse XML files.

Keywords: Word Segmentation, Information Extraction, Text Classification, SVM Algorithm, Random Forest Algorithm, Analysis.

1. Introduction
With the advent and rapid development of the data age, the term "big data" is being mentioned more and more. People use it to describe and define the massive data generated in the information explosion era and name the related technological development and innovation. In this era of data explosion, "data is business opportunity", and various fields have begun to quantify. The development of science and technology leads to the change and advancement of the times. The business development of modern enterprises is closely related to data. Many data analysts use systematic data analysis and data mining methods in the enterprise to analyze and research large amounts of data, and finally obtain results. To support corporate decision-making and business operations. Speaking of the amount of data, the most worth mentioning is the encyclopedia website Wikipedia, which is a free and open website for the whole world. Wikipedia is an online encyclopedia project, characterized by free content and free editing. It is currently the largest and most popular reference book on the global Internet, and it ranks among the top ten most popular websites in the world.
2. Related works
Researches on Wikipedia are different at home and abroad. In terms of the overall situation, foreign research is more diversified, mainly manifested in the diversification of research perspectives, the diversification of research methods, and the continuous updating and progress of research technology. In contrast to the current situation of domestic research, the scope and depth of domestic research are only limited to the fields of libraries, pedagogy publishing houses, and so on. In addition, the domestic research starting point is lower than that of foreign countries, and the time is shorter. Due to the technical level and language constraints, the domestic research level is generally lower than the foreign research level. However, according to relevant data in recent years, research on Wikipedia at home and abroad has followed a similar development trajectory. Domestic attention to the Wikipedia website began in 2005 with an "Encyclopedia in the Internet Age". In this book, Wikipedia technology, the principle of neutrality written by Wikipedia, and the problems in its development were introduced. With the increasing popularity of Wikipedia in China, more and more scholars are studying the encyclopedia from multiple angles. It is mainly reflected in the following aspects: (1) Wikipedia's new knowledge production and sharing model has received extensive attention from domestic scholars, who believe that "Wikipedia is the best representative of knowledge sharing, and the best interpretation of the most free and shared knowledge". (2) In terms of item quality control, most domestic scientists use qualitative description research to analyze the quality control process of Wikipedia. (3) In the application of Wikipedia, the current domestic scholars' research is limited to the fields of library science, pedagogy, and publishing, and it is in its infancy.

3. Network super large text database download

3.1. Project flow chart introduction
Before doing this experiment, we must first determine what data we need to search and download? In this article introduction, we mainly want to download the latest article in January 2019. After determining what to download, we will start downloading data. Since most of the text downloaded on the Wikipedia website is forbidden to download, we choose to pass the Program to download data. After downloading the data, we need to parse the data and use the SAX parsing library and mwparserfromhell interface to parse the data. In the process of analyzing the data, the article category information is extracted with the help of the information template box of Wikipedia, and then word segmentation is performed according to the grammar, stop words are removed, and the data is represented by vectors. Finally, two classification algorithms SVM and model training are used, and the best algorithm is selected for article classification, and the final conclusion is reached. The flow chart is as follows:
3.2. **Programming search database**

First, we request a Wikipedia dump from Wikipedia, the database dump file, or the file named `*-pages-articles.xml.bz2`, which is updated approximately once a week. This file contains the current version's entries, templates, picture descriptions, and basic meta pages (not including discussion pages and user pages)\[2\]. We will search for enwiki which contains dumps of English from Wikipedia. The first request finds the most recent dumps available and lists them (a dump is a snapshot of all existing information in the database).

The code implementation is as follows: List the latest updated article information in the Wikipedia dump file. The result of the following figure shows that the latest updated article dump list of Wikipedia since 2019 is divided into January, February, March, and April dumps.
3.3. Programming download library

First, find the linked HTML file of the relevant URL, parse the HTML data before downloading the data, use the class file, write a for loop to traverse all "li" elements, because the file name is too cumbersome, so use the split() method to cut the file name, only keep part of the file name of enwiki + 20190320 + articles + multistream.xml.bz2.

```
Fig.2 Article Dump List
```

```
In [6]: root_dump = BeautifulSoup(dump_html, 'html.parser')
files = []

for file in root_dump.find_all('li', {'class': 'file'}):
    text = file.text
    if 'pages-articles' in text:
        files.append((text.split() [0], text.split() [1:]))

files[:8]
```

```
Out[6]: [('enwiki-20190320-pages-articles-multistream.xml.bz2', ['15.8', 'GB'] ), ('enwiki-20190320-pages-articles-multistream-index.txt.bz2', ['201.2', 'MB'] ), ('enwiki-20190320-pages-articles-multistream1.xml-p10p30302.bz2', ['163.5', 'MB'] ), ('enwiki-20190320-pages-articles-multistream-index1.txt-p10p30302.bz2', ['163.5', 'MB'] ), ('enwiki-20190320-pages-articles-multistream2.xml-p30304p88444.bz2', ['201.4', 'MB'] ), ('enwiki-20190320-pages-articles-multistream-index2.txt-p30304p88444.bz2', ['201.4', 'MB'] ), ('enwiki-20190320-pages-articles-multistream3.xml-p88444p200507.bz2', ['260.7', 'MB'] ), ('enwiki-20190320-pages-articles-multistream-index3.txt-p88444p200507.bz2', ['260.7', 'MB'] )]
```

```
In [7]: files_to_download = [file[0] for file in files if '.xml' in file[0]]
files_to_download[:5]
```

```
Out[7]: ['enwiki-20190320-pages-articles-multistream.xml-p10p30302.bz2', 'enwiki-20190320-pages-articles-multistream1.xml-p10p30302.bz2', 'enwiki-20190320-pages-articles-multistream-index.txt-p10p30302.bz2', 'enwiki-20190320-pages-articles-multistream2.xml-p30304p88444.bz2', 'enwiki-20190320-pages-articles-multistream-index1.txt-p30304p88444.bz2']
```

```
Fig.3 Analytical Data
```
The above code obtained the download link of all the data on Wikipedia that day through the analysis of this webpage. In traversing all the webpage links, the screening work was carried out, leaving only the enwiki-20190320-pages-article-multistream.xml.bz2 file, and filtering out other files. After determining the download target, the next step will be to download the required data programmatically. In this part, the python urllib library will be called to implement the download[3].

Call python's matplotlib drawing tool to draw a bar graph of article size to view the overall data size. The data compression package downloaded from the webpage through programming varies in size, and the total data is 15.8 GB. In order to more vividly check the data comparison and whether the data download is damaged, the next step is to write a visual code to count all the data. And display it with a bar graph. The code is implemented as follows:

![Fig.4 Data Size Bar Graph](image.png)

4. Analysis of network hyperscale text database

4.1. SAX analysis

These files downloaded in the above process accounted for a total of 15.8 GB, decompressed into an xml file, which is close to 50 GB. We can decompress each file into XML and then parse it. First we need to process one of the files, and then develop a function that can run on all files. First traverse the compressed file. When traversing the compressed file, you need to use the system's bz2 utility, which decompresses the bz2 compressed file and sends the content to the standard output. In this project, we use the bz2 method to decompress. Select one of the data compression packages to decompress, and decompress the bz compressed file into an XML file. The decompressed data shows how many text articles, article titles, users, article content, and other information contained in the compressed package. Take the 8th compressed package file as an example: there are 25104300 lines of text in the compressed package.
After decompression, the XML file is obtained, and then the SAX parser that comes with python is called to parse the XML file, parse the file, and find 492304 article page information.

4.2. mwparserfromhell analysis
The above process realizes that the SAX parser parses the decompressed file and parses it into an XML file. The next step is to process the actual text of the article. At this stage, python's mwparserfromhell library will be used. MediaWiki is used for Wikipedia and many other data projects and provides a relatively standardized template to create wiki pages. Because of this standardization, we can use a custom-built parser to browse the article, and the text of the article itself is processed using mwparserfromhell. The following code is to convert the parsed text content of the article into wikicode, which will extract the classification information of the article later, and continue...
to parse the XML file, extract the classification information after parsing, and prepare for word segmentation and algorithm selection.

![Fig. 7 Convert Text Content into Wikicode](image)

5. Network ultra-large-scale text classification

In the data preprocessing stage, word segmentation technology will be used to segment Wikipedia text articles. In modern text mining technology, there are two commonly used word segmentation methods: Chinese word segmentation and English word segmentation. Chinese word segmentation is more complicated than the English word segmentation. Simple English word segmentation does not require any tools. Words can be segmented through spaces and punctuation. For further English word segmentation, you can use the nltk natural language processing toolkit. For the Chinese word segmentation, jieba is generally used. The jieba tool library optimizes and improves word segmentation tools based on the characteristics of the Chinese language, which is more suitable for Chinese word segmentation tasks.

Text classification is one of the most widely studied tasks in natural language processing, and automatic classification of text content is realized by constructing a model. The general process of text classification: text preprocessing, extracting text features, and constructing a classifier. Among them, the most researched is text feature extraction, which is more broadly text representation. There are three types of text representation models: vector space models, topic models, and neural network-based methods. This project is based on the vector space model to achieve text feature extraction. The vector space model represents the text as a vector composed of real-valued components. Generally
speaking, each component corresponds to a term, which is equivalent to representing the text as one in the space point.

6. Conclusion
In this project, data analysis and text classification were mainly completed. First, find the dump of Wikipedia, find the latest text data in 2019 from the dump file, perform HTML parsing, and then programmatically download the data and decompress the data. During the download process, the data of dumped pages can be divided into many forms, but in this experimental project only the data of -pages-articles.xml.bz2 is needed. After the file is downloaded, the analysis toolkit is called to analyze data, cut data, and complete data information extraction and data classification. Finally, SVM algorithm and RandomForest algorithm are used for model training, and the actual effects of the two classification algorithms are compared. Finally, the optimal algorithm is selected to realize the automatic classification of text articles[4]. The significance of this project is to provide a convenient and efficient text download, text analysis, and text classification method for data science projects, reduce the time cost of data download, and improve the efficiency of data processing and analysis. It allows programmers to directly download, parse, and automatically classify Wikipedia text articles through the program, which improves an efficient and convenient way for data source collection and data preprocessing. There are several types of techniques used in the experimental application process: find and download data programmatically from the Internet, use Beautiful Soup to parse HTML, use SAX to parse XML, use mwparserfromhell to parse Mediawiki, and data preprocessing (word segmentation, remove stop words) classification algorithm model training.

Acknowledgement
This work was supported in part by the National Natural Science Foundation of China, grant number 72073041. Open Foundation for the University Innovation Platform in the Hunan Province, grant number 18K1032, 011 Collaborative Innovation Center for Development and Utilization of Finance and Economics Big Data Property, Universities of Hunan Province. Open project, grant number 20181901CRP03, 20181901CRP04, 20181901CRP05. 2020 Hunan Provincial Higher Education Teaching Reform Research Project under Grant HNJG-2020-1124, HNJG-2020-1130. 2020 General Project of Hunan Social Science Fund under Grant 20B16.

References
[1] Bingying D Research and implementation of web crawlers in search engines (2014 the academic degree's dissertation of Beijing University of Posts and Telecommunications)
[2] Zhaotao Z, Dongbo B A Preliminary Study on Graphical Representation of Texts J. Chinese Journal of Information (Vol 19 Iss 2)
[3] Huosong X, Baoguo L Dynamic 2016 web page evaluation crawler algorithm based on Python J. Software Engineering (Vol 2)
[4] Jinshu S, Bofeng Z and Xin X 2006 Research Progress of Text Classification Technology Based on Machine Learning J. Journal of Software 17 pp1848-1859