The Analysis of Web Page Information Processing Based on Natural Language Processing

Yusheng Zhao [a]

International School, Beijing University of Posts and Telecommunications, Beijing 100876, China.

[a]zhaoyusheng@bupt.edu.cn

Abstract. Nowadays, the structure of webpages has gradually become more and more complicated, and the content of webpages has gradually increased. This has caused a lot of useless and even illegal information in webpages. The screening of keywords in webpage information and the evasion of invalid illegal information have become the focus of attention. This paper will use natural language processing (NLP) technology to crawl web page information and then process it, in order to avoid some invalid or illegal information, and to find out the key information in the web page. Therefore, this paper also concludes that NLP is reasonable and practical for applications on web pages.

Keywords: Natural Language Processing, Python, Crawler, Word Segmentation, TF-IDF.

1. Introduction

The increase in the amount of web page information has caused the complexity of the web page structure. A lot of questions come along: how to quickly find keywords in a web page, how to avoid invalid or illegal information in a web page, and so on. Currently, NLP is one of the more common solutions.

Natural language processing is a field of research and application. Its main purpose is to study how to make computers understand natural language text or speech and manipulate them to do something meaningful [1]. The way humans understand and use natural language will be collected by NLP developers and used to develop appropriate tools and techniques to enable computer systems to understand and manipulate natural language to perform the tasks they need. There are many basic disciplines in NLP, including computer and information science, linguistics, mathematics, and psychology [2]. NLP is widely used, including many areas: intelligent translation, text processing, speech recognition and more. This paper will focus on text processing for discussion and analysis.

This paper will solve the problem of finding keywords and avoiding illegal and invalid information through an example program, and discuss and research the development prospects of NLP. This paper will first introduce NLP from a more macroscopic and holistic perspective. Then this paper applies specific experiments to analyze the text processing in the traditional NLP field. Finally, this paper will compare and analyze the related algorithms of text processing in the NLP field. This paper is mainly based on the experimental algorithm around the structure.

2. Literature Review

Natural language processing began in the 1950s and has undergone several major changes in the process of development: for example, "conceptual ontologies" that emerged in the 1970s, turning the world's information into computer-understandable data; and in the 1980s, The introduction of language-processed machine learning algorithms has revolutionized the way most natural language processing systems are based on complex handwritten rules. The reference to unsupervised and semi-supervised learning algorithms is also due to the over quantization of data volume [3].

At present, the main application technology of NLP is divided into the following four aspects: Syntax, Semantics, Discourse and Speech. In the Syntax section, techniques such as morphological restoration, morphological segmentation, and part-of-speech tagging will solve the problem of grammar involved in NLP. In the Semitics section, techniques such as named entity recognition (NER), natural language understanding, and natural language generation will solve the semantic problems involved in NLP [4]. In the Discourse section, Automatic summarization, Coreference resolution, and Discourse analysis techniques can discover the intrinsic links between statements or between vocabularies. Finally, in the Speech section, speech recognition and speech segmentation techniques will be a good tool for solving natural speech analysis problems [5]. Among the above four points, Syntax, Semantics will be the main application technology of this paper.

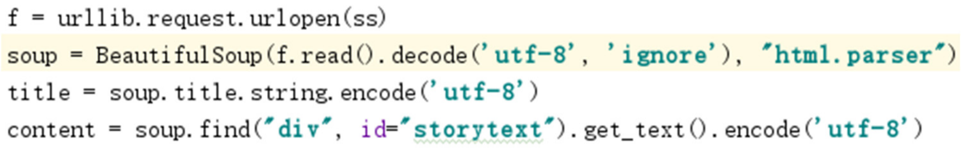
This paper will analyze and discuss NPL technology through practical applications. The specific method is divided into four steps: First, the python code crawler is used to crawl the specified piece of information of a news website. Subsequently, the repeated data is removed by a deduplication algorithm, which mainly applies the md5 compression algorithm in hashlib. Next, all the resulting data is processed so that it is stored in the CSV file. Finally, the main information processing is done for tabular text. This approach applies most of the techniques in the Syntax and Semantics sections of NLP technology and can be used as a basis for discussion and analysis.

3. Architecture

This paper will mainly introduce the mainstream algorithms used in this experiment in this section.

3.1 Information Extraction

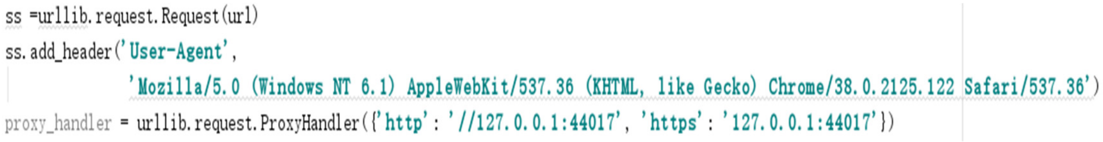
This part is responsible for the classification and crawling of information on the web page. The web pages in the news website selected in this paper are based on HTML pages, so you need to use the browser as an auxiliary tool to open them and extract the div where the main text content of the news is located. In terms of extracting text, this paper mainly uses the beautiful soup library in Python to process HTML web pages. As shown in Figure 1 below.



```
f = urllib.request.urlopen(ss)
soup = BeautifulSoup(f.read().decode('utf-8', 'ignore'), "html.parser")
title = soup.title.string.encode('utf-8')
content = soup.find("div", id="storytext").get_text().encode('utf-8')
```

Fig. 1 Application of the Beautifulsoup library. As shown in the figure, <div> <id=storytext> is identified in the case of this paper.

In order to deal with the rumor mechanism of the target webpage, this paper adopts the method of setting the crawler header. The main principle is to disguise the crawler as the daily access of the browser. As shown in Figure 2 below.

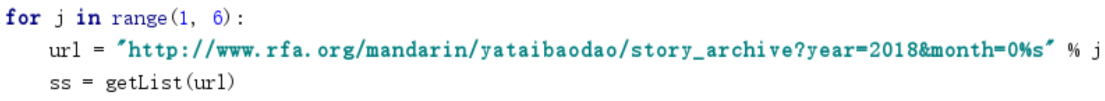


```
ss =urllib.request.Request(url)
ss.add_header('User-Agent',
              'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.122 Safari/537.36')
proxy_handler = urllib.request.ProxyHandler({'http': '//127.0.0.1:44017', 'https': '127.0.0.1:44017'})
```

Fig 2. Setting of the crawler head

It can be seen that in addition to the processing of the header, this paper also sets the IP and port of the entire client computer. The reason for this is to prevent the IP of client computer from being blocked by the visited website.

Finally, this paper analyzes the URL parameters of the webpage and finds that its URL parameters are around time, so the code is adjusted, which makes it easier to set the traversal parameters in the crawler. As shown in Figure 3 below.



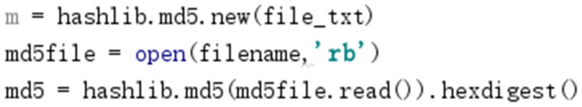
```
for j in range(1, 6):
    url = "http://www.rfa.org/mandarin/yataibaodao/story_archive?year=2018&month=0%s" % j
    ss = getList(url)
```

Fig 3. Crawl URL settings

Finally, the paper downloads the crawled information and stores it in a TXT file that is not wrapped.

3.2 Re-processing

This part re-processes the data crawled by the crawler. Here, this paper mainly uses the md5 value in hsahlib to Duplicate Elimination. The md5 algorithm can be used to effectively compress the file, and finally the file is given a specific md5 value. As long as it is the same file, the md5 value is the same. This paper uses this feature to remove the same file. As shown in Figure 4 below.

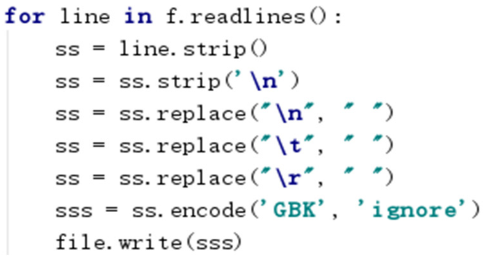


```
m = hashlib.md5.new(file_txt)
md5file = open(filename,'rb')
md5 = hashlib.md5(md5file.read()).hexdigest()
```

Fig 4. Calculation of Md5 values

3.3 Tabulated Data

This part converts and stores all the read data, first stored in the TXT format document, and finally stored in the CSV format file through conversion processing. As shown in Figure 5 below.

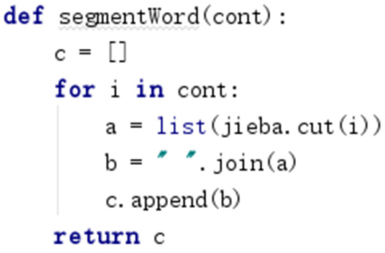


```
for line in f.readlines():
    ss = line.strip()
    ss = ss.strip('\n')
    ss = ss.replace("\n", " ")
    ss = ss.replace("\t", " ")
    ss = ss.replace("\r", " ")
    sss = ss.encode('GBK', 'ignore')
    file.write(sss)
```

Fig 5. File format conversion

3.4 Word Segmentation and TFIDF

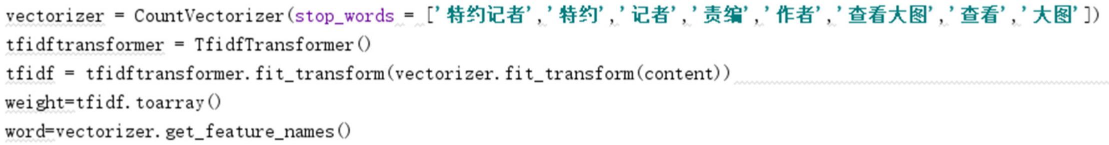
This part is mainly responsible for further processing of the data. For text data that has been converted to CSV format, this paper retrieves the file data and uses the jieba word segment for word segmentation. Here mainly uses the method of calling jieba library in python. As shown in Figure 6 below.



```
def segmentWord(cont):
    c = []
    for i in cont:
        a = list(jieba.cut(i))
        b = " ".join(a)
        c.append(b)
    return c
```

Fig 6. About the setting of word segmentation

For the TF-IDF value after the word segmentation, this paper uses the correlation function in the sklearn. feature_extraction. text library in Python. First generates the word frequency matrix, and then calculates the corresponding weight of each word segment. As shown in Figure 7 below.



```
vectorizer = CountVectorizer(stop_words = ['特约记者','特约','记者','责编','作者','查看大图','查看','大图'])
tfidftransformer = TfidfTransformer()
tfidf = tfidftransformer.fit_transform(vectorizer.fit_transform(content))
weight=tfidf.toarray()
word=vectorizer.get_feature_names()
```

Fig 7. Calculation of TF-IDF. Since some words do not have the meaning of query, this paper sets Stop_words to eliminate its interference with data processing, and its content is important to draw on the Stop_words library of related words in the news. For similar " Special correspondent", " reporter" "The words are set.

4. Discussion of Application and Function

The experimental algorithm applied in this paper performs keyword extraction on target text and avoids invalid information processing. Among them, the beautiful soup library has achieved the expected results for the processing of web pages and for the setting of crawler heads and formats. The use of the hsahlib library is an innovation in this lab. At the same time, the method of judging the corresponding weight of each word segment by using the TFIDF value also achieves the corresponding goal. There are still some areas that can be improved in this paper. For example, the word segmentation algorithm in this paper is a direct call to the jieba library, so for some words whose semantics are not very clear, they are not well segmented. In this respect, ICTCLAS2013 algorithm has great superiority, and its accuracy and speed of Chinese word segmentation reach a very high level, which can solve the problem of unclear part of semantic separation [6]. There is also a need for improvement in the processing of Stop_words. Some illegal words can be added to further eliminate the interference of invalid words or illegal words on the query results.

5. Conclusion

This paper first introduces the history and concepts of natural language processing, and then combines the methods used in this paper to explain the mainstream techniques applied in natural language processing. In the experimental part, this paper mainly uses the specific webpage information to apply the algorithm, and discusses and analyzes the applicability of the applied algorithm. In summary, we have concluded that natural language processing is reasonable and practical for applications on web pages. Of course, this paper is only an application of natural language processing for web pages. In the long run, natural language processing has broad application fields and surprising application prospects. This is a cross-disciplinary subject involving multiple disciplines such as language science, computer science, and mathematics. Its development can advance progress in many areas. It is believed that with the continuous enrichment of data resources of various vocabulary French corpus, the continuous improvement of language analysis technology, the emergence of new methods and new technologies, natural language processing can have better application fields and research prospects in the future.

References

[1]. Chowdhury, G. G. (2003). Natural language processing. Annual review of information science and technology, 37(1), 51-89.

[2]. Manning, C. D., Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.

[3]. Di Marco, A., & Navigli, R. (2013). Clustering and diversifying web search results with graph-based word sense induction. Computational Linguistics, 39(3), 709-754.

[4]. Yucong, D., & Cruz, C. (2011). Formalizing semantic of natural language through conceptualization from existence. International Journal of Innovation, Management and Technology, 2(1), pages-37.

[5]. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations (pp. 55-60).

[6]. Zhang, H. P., Yu, H. K., Xiong, D. Y., & Liu, Q. (2003, July). HHMM-based Chinese lexical analyzer ICTCLAS. In Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17 (pp. 184-187). Association for Computational Linguistics.