Research and Implementation of Key Technology of Distributed Big Data Collection

Guo-song JIANG\(^1\) and Xiao-ling HE\(^2\)

\(^1\)School of Computer Science, Huanggang Normal University, Huanggang, 438000, P. R. China;
\(^2\)School of Journalism and Communication, Huanggang Normal University, Huanggang, 438000, P. R. China

**Keywords**: Big data, Data collection, Text extraction, IP proxy pool.

**Abstract.** With the arrival of big data age, for big data mining and analysis has become a hot research today. The data set is the basis for big data mining and analysis. Therefore, an effective data collection scheme is of great significance to the study of big data mining. An efficient distributed big data collection system is proposed. In this paper, a general and effective text extraction algorithm based on the weight of the label tree node is proposed in the analytic module of the system. At the same time the introduction of IP proxy pool technology to ensure the continuity of the system. Experiments show that the system can efficiently and quickly obtain a large amount of network data, and has strong robustness, feasibility and flexibility.

**Introduction**

With the mobile Internet, e-commerce, social networking and other emerging technologies and application of the Internet, image, video, log and other network data show explosive growth. Taobao nearly 400 million members of the daily production of goods trading data about 20TB, Facebook about 1 billion users daily log data generated more than 300TB [1]. Big data era is already coming, the field of big data has become a popular research topic. The data is the basis for the realization of big data research, the traditional data collection technology program has been difficult to meet the rapid collection of high-quality data set needs. So how to efficiently collect massive high-quality data on the application and research of big data has a very important role.

This paper presents a highly efficient new big data collection technology program. Mainly uses the distributed structure, in the analysis module based on the label tree node weight of the text extraction algorithm. The algorithm can compare the weight of the label tree node, and then cut out the useless information block, which can quickly locate the text information content, eliminating the free information analysis time, improve the speed of analysis. And access to the page for the frequency limit, the introduction of IP proxy pool technology. Through the agent pool into the outflow update mechanism to ensure the availability of agents in the pool. By switching agents to ensure that the system can continue to work, eliminating the waiting time to wait, which will greatly improve the efficiency of the system.

**System Frame Design**

**The Overall Architecture of the System**

The overall framework of the system as shown in Figure 1, mainly includes five modules: **capture module, IP proxy pool module, parsing module, URL processing module and data storage module.** Where the crawl module gets the URL to crawl from the URL queue. And then call from the IP proxy pool to obtain the available agents, from the Internet to capture the original data, and processed by the analysis module. The parsing module first preprocesses the data and removes some noticeable noise. And then extract the text information through the text extraction algorithm based on the weight of the tag tree node. The URL-related data is processed by the URL processing module, and the base data is processed by the data storage module. The URL processing module is mainly used for control of
distributed crawling. While the data storage module is the data regularization and persistence, for the follow-up analysis and processing to lay the foundation.

**Distributed Architecture**

The system uses the master / slave distributed architecture. As shown in Figure 2, the master control node extracts the URL from the URL queue to be crawled to each crawler. And then by the capture host to complete the collection task and resolution tasks and has been successfully crawled the URL and the new URL extracted to the main control node. The URLs that are successfully crawled are cached in the crawled collection, and the new URLs are filtered based on the crawled collection and cached to the corresponding queue to be crawled. Where the crawl queue and the crawled collection are implemented by using the memory database redis. The queue to be crawled takes the pre-allocated policy for subsequent crawling.

**Text Information Extraction**

About the text information extraction, domestic and foreign scholars have done a lot of work research. The literature [2-5] uses an algorithm based on visual features. This algorithm is based on the VIPS (vision-based page segmentation) web page block algorithm proposed by Microsoft Research Institute. Although it can achieve very good results, but the VIPS algorithm is relatively complex, and the number of iterations, while it also depends on the browser kernel code, it takes a long time. And the literature [6] according to the similarity of the classification of the page, for each type of web training to get the template, and then according to the template to extract the unknown page. The method cannot be applied to the structure of different web page text extraction. In [7], the method based on the mark window is used. Mark the window of the reference is good, but for each mark
window text must first word, and then calculate the word sequence distance. Not only on the word segmentation technology has a higher demand, but also there is no problem of efficiency.

Based on the generality and efficiency of the above algorithm, this paper presents an efficient and efficient text extraction algorithm based on the weight of tag tree nodes. It blocks the pages based on specific tags and builds the tag tree. And then calculate the weight of the label tree node from the bottom up, by comparing the weight, pruning, and ultimately retain the text of the subtree. Experiments show that the algorithm can extract the text information in a short time, so as to improve the efficiency of the analysis.

**Construct the Tag Tree**

At present, the main block labels of most of the page layout are `<div>`, `<table>`, `<p>`, so only use these tags to build the tag tree. In the construction of the label tree, but also on the source HTML document to do the first pretreatment, remove some obvious noise. Such as the internal style text `<style>` block in the document, the header tag `<head>` block, the JavaScript script `<script>` block, and the comment `<!- ... ->` block. This article uses regular expressions to remove these noises.

When constructing a tag tree, use the stack as a secondary space. The specific construction steps are as follows:

1. The root label `<HTML>` into the stack.
2. When the block start tag is encountered, the block is used as the child node of the stack node. And put the block into the stack.
3. When the end of the block is encountered, the stack is ejected.
4. If the stack is empty, the structure is over. Otherwise continue scanning, encounter block start tag jump (2), encounter block end tag jump (3).

Specific structure shown in Figure 3. Where (a) is the HTML source document. A-G is a block label. And (b) is the tag tree constructed by (a).

![Figure 3. Tag tree construction.](image)

**Text Information Extraction Algorithm**

Because the text information is the most part of the content, therefore, in general, most of the pages after the block processing, the text block always contains the sub-tree which has largest number of blocks, and The block numbers of the sub-tree is quite different from others brother sub-tree in the same layer. Therefore, this paper proposes a text extraction algorithm based on the weight of tag tree nodes. On the basis of constructing the tag tree, the weights are assigned to each block node from the bottom up. The assignment rules are:

If the node is a leaf node, its weight \( w_i = 1 \).

If the node is a non-leaf node and \( N \) is the set of all the child nodes of the node, its weight \( w_i = \sum_{j \in N} w_j \).
(1) While assigning a block node, it is also to compare this node with the node with largest weight among the sibling nodes. Let node $i$ have the weight $w_i$, and the maximum weight of its sibling node is $w_{\text{max}}$. Let the variable $R$ is $R = \frac{w_i}{w_{\text{max}}}$. 

(2) When the variable $R < Q$, then pruning the node $i$. Where $Q$ is the threshold set by experience. Figure 4 shows the specific pruning when the threshold $Q = 4$. Obviously, by pruning, you can reduce the analysis of five useless blocks. Only need to contain the text of the nine nodes to analyze just fine. Therefore, we improve the resolution efficiency by 35.7% through the text extraction algorithm. So when the useful information block nodes are relatively more time, then the resolution efficiency will have a greater increase. Can be seen by calling the text extraction algorithm, continue to prune, you can remove the ads, navigation links, copyright information and other useless information nodes, so as to get the final text information node. This will reduce the resolution of useless information, so there is no need to waste time to resolve useless information, greatly speeding up the web analytics.

**IP Proxy Pool**

Some sites in order to protect their own data security or reduce the burden on the site, its access to the same IP will be limited. When an IP access request exceeds its reach, the site will take an interception that prevents the IP from accessing the site and will release the IP limit for a certain period of time. To solve the IP limit problem, this article uses the IP proxy pool mechanism. That is, using a common proxy pool to switch agents. Each time a certain amount of data is fetched or after a certain period of time, or when an IP restriction occurs, a new available agent is obtained from the proxy pool and the access is continued to the agent.

The agent inside the proxy pool is not fixed, it will continue to get available agents from the relevant proxy page into the proxy pool. And will automatically check the agent on the pool to update, and some useless proxy to delete. As shown in Figure 5, each agent in the pool has a flag that records the status of the agent: useful, useless, and useless. Each time a proxy labeled as useful is extracted from the pool, the agent is assigned to the crawler and the agent is modified as using. If the proxy is marked as useless from the crawler because of IP restrictions, the normal return flag is useful. When the trigger is automatically updated, all agents marked as non uses are detected in the pool. After the detection of the useless agents are removed from the pool.

The use of IP proxy pool mechanism can try to avoid IP restrictions occur, if the IP limit occurs, it can also save the time to wait for IP limit release, which ensures the success of the crawler host operation, thus ensuring the amount of data to crawl. Similarly, when some need to log in to visit the site, such as microblogging, can join a similar user account pool, regular switching users to access.
Experimental Analysis

In this paper, Sina microblogging as an example to verify the feasibility of the program. The experimental cluster machine CPU is Intel (R) Xeon (R) 2.60GHz, memory is 2GB, the operating system is centos 6.5. Which uses a machine to collect user information, a machine to collect microblogging information, two machines to collect comments. Procedures with java development, packaged into jar into the machine cluster to run. The cluster continues to run for 12 hours.

The amount of data collected is shown in Table 1. According to the data of Table 1 and the literature [8] results were compared to Figure 6a, Figure 6b. It can be seen that the scheme is based on the improved rate and the continuous stability of the literature [8]. Mainly because this paper presents a text extraction algorithm based on the weight of the tag tree nodes, which greatly reduces the useless non-subject information blocks, thus speeding up the resolution rate. The introduction of IP proxy pool technology to ensure the continuity and stability of the system.

Table 1. The amount of data collected by the cluster within 12 hours.

| User Info | Microblogging | Comments information |
|-----------|---------------|----------------------|
| 508751    | 714562        | 1539835              |

Figure 6. Comparison of experimental results.

To verify the scalability of the system, we will increase the number of machines collecting comments from 2 to 4. The average weekly collection of comments increased from 128,300 to 23.54 million. Although the performance is only 1.8 times to enhance, and did not achieve the ideal 2 times the upgrade, the reason may be due to bandwidth or network reasons. But also shows that the system has good scalability. Therefore, in the case of bandwidth support, can be a simple increase in the collection of machines can be a corresponding increase in the amount of data.
Conclusion

In this paper, an efficient big data collection scheme is proposed, and a text extraction algorithm based on the weight of the tag tree node is proposed in the analytic module. The algorithm can eliminate the useless non-text block, which improves the efficiency of the analysis. And the introduction of proxy pool technology for IP restrictions to ensure the continuity and stability of the system. The scheme is based on a parallel distributed crawl method with strong scalability. In the case of bandwidth allows, by simply increasing the crawl node can increase the amount of data collected. Experiments have proved that the program has strong usability, for the follow-up big data research has laid a strong foundation.

Acknowledgment

This work is supported by Research project of Hubei Provincial Education Department (No.D20152903, 15Y159), Foundation of Huanggang normal university (No. xfg2015A11, 2014015103, 201616303), Thanks to the reviewers for the valuable comments helping to improve the quality of the manuscript.

References

[1] Li Bin. Big data and its development trend [J]. Guangxi Education 2013, 35.
[2] K Simon, G Lausen. ViPER: augmenting automatic information extraction with visual perceptions[C]. Proceedings of the 14th ACM International Conference on Information and Knowledge Management, 2005: 381-388.
[3] Chibane L., Doan B-L. A web page topic segmentation algorithm based on visual criteria and content layout[C]. SIGIR, ACM, 2007.
[4] Huang Wenbei, Yang Jing, Gu Junzhong. Study on the algorithm of web page text extraction based on chunking [J]. Journal of Computer Applications. 2007, 27 (6): 24-26.
[5] An Zengwen, Xu Jiefeng. Research on Web page text extraction method based on visual feature [J]. Microcomputer and application. 2010, 9 (3): 38-41.
[6] Huang Wenbei, Yang Jing, Gu Junzhong. Study on the algorithm of web page text extraction based on chunking [J]. Journal of Computer Applications. 2007, 27 (6): 24-26.
[7] Zhao Xinxin, Suo Hongguang, Liu Yushu. Technology of web page text extraction based on tag window [J]. Application Research of Computers. 2007, 24 (3): 144-145.
[8] Sun Qingyun, Wang Junfeng, Zhao Zongqu, etc. A Data Acquisition Scheme Based on Simulated Login [J]. Computer Technology and Development, 2014, 24 (3): 6-10.