DESIGN AND IMPLEMENTATION OF CARPOOL DATA ACQUISITION PROGRAM BASED ON WEB CRAWLER

Junlei Zhao and Bicheng Li

Cyprus College of Computer Science and Technology,
Huaqiao University, Xiamen, China

ABSTRACT

Now the public traffics make the life more and more convenient. The amount of vehicles in large or medium sized cities is also in the rapid growth. In order to take full advantage of social resources and protect the environment, regional end-to-end public transport services are established by analyzing online travel data. The usage of computer programs for processing of the web page is necessary for accessing to a large number of the carpool data. In the paper, web crawlers are designed to capture the travel data from several large service sites. In order to maximize the access to traffic data, a breadth-first algorithm is used. The carpool data will be saved in a structured form. Additionally, the paper has provided a convenient method of data collecting to the program.

KEYWORDS

Web-crawler, carpool-data, structured;

1. INTRODUCTION

Vigorously developing public transport is a development strategy of China in the field of urban transportation. In recent years, the urban public transport systems has become large-scale and the overall level of services is significantly improved. However, the diversification of travel demand between large or medium-sized cities is becoming more and more obvious. The coverage of public transport hub networks are limited. The problems of the conventional public transport service models are more and more prominent. The contradiction between the diversificated demands of public transports and the status of public service is becoming more and more obvious [1]. At present, many scholars have been exploring the application of data mining in traffic traveling. How to obtain these carpool data has become a difficult problem. There are some data acquisition methods. The first method is to collect the traffic data from bus systems of companies. The second method is to collect the traffic data of cities manually. The third method is to access the online traffic data through the web crawler. The cost of the first approach is costly; the second one is difficult to be implemented. Therefore, the third method is used to design a dedicated web crawler for a particular site to collect data. After 20 years of rapid development of network technology, the original Internet is undergoing earth-shaking changes **.** Internet pages grew from thousands of pages in 1993 to more than 2 billion pages at present [2]. Many different types of data are being served on the internet. In order to download these data, search engine programs need to be developed based on web crawlers. The search engines obtain large-scale hyperlinks with the help of the Web crawlers and store the downloaded web pages in large databases and provide indexes interfaces for user queries. A web page is the basic structure of some basic script

DOI:10.14810/ijmech.2018.7401

labels. The text information is typically stored in the <u> label and may be in <a> <na> or <p> <np>. But in some pages the link and text information are together, it is necessary to separate them and save only the required information. Web crawlers are programs, which fetche information from the World Wide Web in an automatic manner. Web crawling is an important search technique. The crawler is a search engine component. It accesses portions of the Web tree based on certain policies and collects the retrieved objects in the local repository. The process contains three steps, (1). Downloading Web pages. (2). Parsing through the downloaded pages and retrieving all the links. (3). For each link retrieved, repeating the process above. Through the above three steps of the web crawler, data can be downloaded from some simple web pages. For those pages, that are more complex or have protective measures, web crawlers need to be designed based on specific structures.

1.1. Research Status of Web Crawlers

In the study of web crawlers, the research status will be introduced by both distributed and non-distributed web crawlers. For distributed web crawler, Sawroop Kaur Bal et al. [3] discussed the design of the Smart distributed web crawler systems. Good crawling strategy determined which page should be downloaded. The distributed web crawler [4] aims to evenly divide the workload of a regular crawler onto multiple nodes. The number of nodes were arbitrary, the presence of nodes were dynamic, individual nodes were able to change the status of them back and forth between absent and present in the environment at any time. UbiCrawler [5] is presented, as a fully distributed, scalable and fault tolerant Web crawler. UbiCrawler introduced new ideas in parallel crawling, in particular the usage of consistent hashing as a means to decentralize completely the coordination logic, for graceful degradation in the presence of faults and linear scalability. A crawler [6] is designed based on Peer-to-Peer networks. The distributed crawler harnesses the excess bandwidth and computing resources of nodes in systems to crawl web pages. Each crawler was deployed in a computing node of P2P to analyze web pages and generate indices. The control node was in charge of distributing URLs to balance the load of the crawler. A distributed web crawler [7] is created for crawling the information of goods on the e-commerce sites. Customers could use the information as a reference to buy something on the internet. Also, a distributed system crawling on the server was faster compared to the crawling did by the master system alone. The distributed network crawler greatly improved the efficiency of large data processing, but also make full use of resources. However, according to the needs of the project, this article only needs to crawl the carpool data of one City and do not considerate a distributed architecture. For non-distributed web crawlers, a method [8] was developed that uses link analysis to determine what constitutes a good content analysis metric. The routing information encoding into backlinks also improved topical crawling. A web crawler [9] is developed that the process of crawling, once a web page is downloaded, they parse the DOM tree of pages after preprocessing (eliminating stop words and stemming) and then the page will be classified by a conventional classifier with a high dimensional dictionary. For webpage mining [10], in this paper explore the role of web crawlers in webpage mining and explore how to construct a theoretical web crawler framework for web mining. LinCrawler[11] is implemented their semantic topic-specifc crawler using Lin semantic similarity measurement and the relationships of WordNet concepts. Experimental results proved the superiority of LinCrawler over TFCrawler (the crawler works lexically) and BFCrawler (the Breadth-First crawler). WCSMS [12] is developed a query system, which collected sale data from different online malls by using web crawlers. Users can then access the data and browse the data on the webpages of the WCSMS. Meanwhile, this sale management system can be used to solve the inventory problem of physical malls. Scholars in related fields have conducted some research.

2. THE CHARACTERISTICS OF WEB CRAWLER

The web crawler in the paper uses breadth-first algorithm to crawl multiple page links and gets as many links as possible. Some methods are also employed to clean the list of URLs. Be sure to get the correctness of the target link and data**.**

2.1. Description of the Problem

As the carpool websites is a huge, widely distributed, highly heterogeneous, semi-structured and highly dynamic information warehouse, the excavation of Web information resources can not directly use the traditional data mining technology. New data models, architectures and algorithms are needed. On the basis of researches in this field, web crawler needs to overcome problems, which are the data format conversion and content unstructured conversion.

2.2. Initialization of URL

First of all, to crawl work, some seed URLs must be artificially initialized. Whether from the implementation of difficulty ,or from the point of view, the breadth-first search strategy, is a good choice. So in the article, the initial URL are taken as the entry, and then the crawler downloads the URLs to the URL list. Web crawlers will be stopped when all URls are downloaded or when a certain condition is satisfied. In order to download the URLs completely, a loop is needed to judge whether the next page has URLs. After the URLs in current pages are downloaded successfully, the URLs in the next page will be downloaded by web crawler. The extracted URLs are stored in a URL list. Then URLs list will be accessed to detect whether there are duplicate address links. In the acquisition phase, the crawler mainly communicates with the Web server through the http protocol, and uses the Http Web Request to request and respond to the corresponding server through HEAD, GET and POST.

2.3. Page Request

In the page request phase, the crawler gets the information from servers, but the information obtained is byte stream. In order to convert it into text form, the encoding language of web pages must be identified. Therefore, it is necessary to obtain the coding type of web pages. Most of web pages are encoded with GB2312 or UTF-8 and the coding type of web pages can be gotten from the <script .... charset=”GB2312”><n script> or <meta charset=”UTF-8”><n meta>. For the porpuse of obtaining the coding type of the web pages, a regular expression(@ ”<.*? [n s n s] +? Charset = [ns] [,,,,]? (.?)”” [nS ns]*?>”) is employed to extract charset. In addition, page information of some websites, such as sohu.com, is so large that web pages are transmitted in the gzip compression format. For such pages, we have to determine whether it has been compressed by gzip before extracting the code. If so, they are decompressed. The page information can be converted from a byte stream to a string when the charset is gotten. The regular expression is used to get the coding type of the carpool websites and the charset is UTF-8. In next step, the downloaded web pages will be analyzed.

2.4. Page Analysis

More than three methods, such as “beautifulSoup”, “regular expression” and “Lxml”, can be used to parse the page. BeautifulSoup library is employed to manipulate html documents and extract information as following: (1) text information, the find () function is used to get the text information from the html document through the< class =”....” > tags. (2) url links, tags which contain URLs links are < a href =”....” > < iframe src =”....” > < base href’.. ’ > < img src = ””

Body background = "...." > and so on. Sometime, URLs in the web pages use the relative address method, so the URLs need to be converted from a relative address to an absolute address. The usual crawler as long as the following three points can be extracted. If some specific pages are needed with more detailed requirements, the page can be analyzed in further details. Figure 1 shows the html page of carpool web sites, where the URLs in the module< a href ="...." > and the text information need to be extracted in the module<class = "view ad-meta2">. So the find all () function is used to find the attributed nodes of URLs, and the find () function is employed to get the text content of carpool data.

2.5. Page Storage

Usually, web pages are downloaded by web crawlers and saved as following forms. (1) The binary stream files are stored directly on the local disk without any processed; (2) The form of html of text file is stored in the local disk; (3) Plain text content, whose tags are removed by either the find() or the find all() function. In our way, web pages are downloaded and done other operations in plain text content. And then, the crawler completes the job and continues reading the next URL in the URL list.

2.6. Theoretical Analysis and Comparison

After the study above, the crawler is theoretically feasible. By introducing the web crawlers designed in section one, it is found that when users have specific requirements, they need to design web crawlers for web structures and contents. Depending on the project's needs, some of the measures can be used to process the data simply during the collection process. In the context of the project, the use of web crawlers to capture data is operational. Web crawlers can provide data supporting for the project.

3. THE PROGRAM ARCHITECTURE

The designing web crawler can obtain and retrieve data among web pages. On the whole, the data can be accurately obtained by the web crawler so that it can satisfy the needs of route planning, seat reservation and timing. From the local point of view, the core algorithm of each step is designed in accordance with the specific attributes of the web pages. And they are in close cooperation among them. A detailed description of the web crawler method and the handling of the details will be given now. First of all, Figure 1 shows the flow chart of web crawler.

Through the flow chart, it can be clearly illustrated how to get the carpool data. The Web crawler uses the page resolution method to get the Beautiful-Soup library function. By calling the find(), find all() and other functions, the required data information can be easily obtained. The following is the analysis of the find() and find all() functions. The find all() method looks through a descendant of the tag and retrieves all descendants that match filters. The find all() method scans the entire document to search results, but sometimes it only needs to find one result. If the document has only one body tag, scanning the entire document for more content is a waste of time. Since most web pages do not have a good HTML format, BeautifulSoup needs to determine its actual format. Figure 2 shows the workflow. Here the program will be introduced step by step. In order to clearly present the whole process, the program is divided into three parts.

.

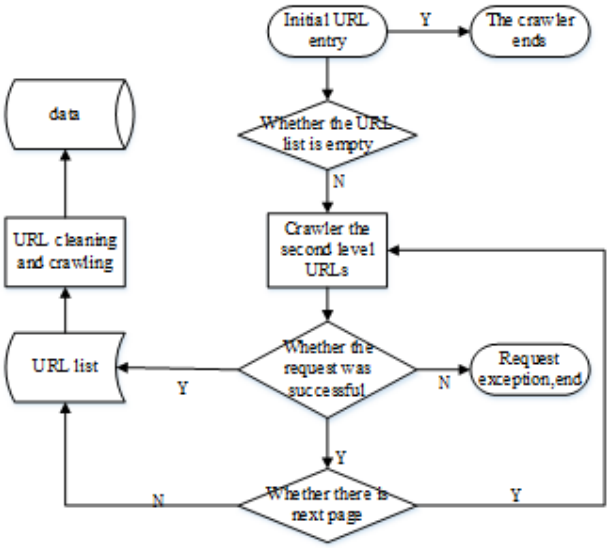


Figure 1. Web Crawler Process

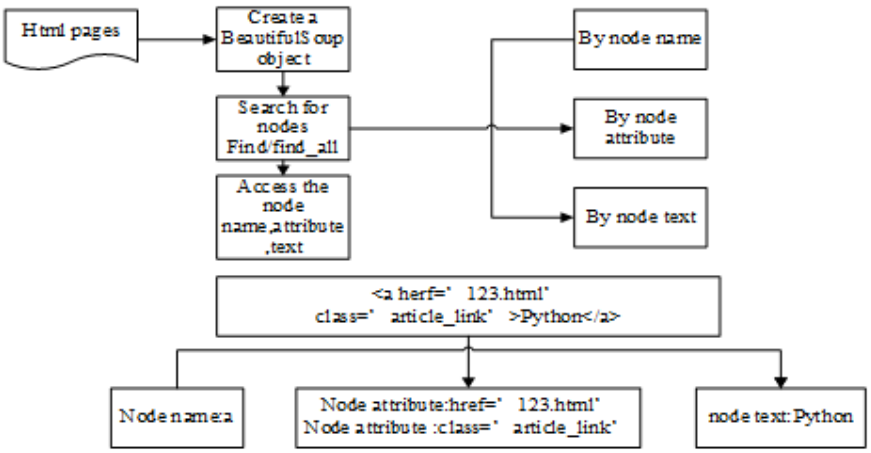


Figure 2. The Processing Of BeautifulSoup

3.1. Crawling URLs

Setting the initial URLs as the entry, the URL links of second page are obtained from the first page. Due to the specific structure of the carpool web sites, breadth-first search strategy is used to get and save all the URLs to the list. The breadth-first search strategy means that the web crawler completes the current page during the crawling process and then searches for the next page. The unstructured or semi-structured nature of the sites make it possible to have a lot of duplicate data and advertising links in the crawled links, and cause interference to the subsequent data acquisition, so it is required to do the second step.

3.2. URL Cleaning

The URLs are filtered in the list one by one to remove duplicate links. The number of characters is the same in each second page link, while the advertising links and other links are different. The

feature is used to filter out these non-data link operations and remove the noisy links and then update the URL list.

3.3. Crawling Carpool Data

After getting the cleaned URL list, the previously described BeautifulSoup library is utilized to search the carpool data modules, and then the find() function is used to download the data. Figure 3 shows the algorithm flowchart.

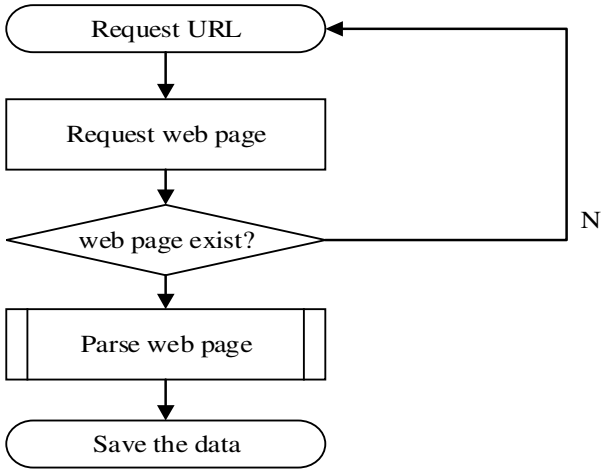


Figure 3. Algorithm Flowchart

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1. Experimental Results

From the first URLs, the find() method is used for the web crawler to get the tag <a href ,..URLs..,> and download the URLs of the second web page. Then put them into the list, and get rid of the URLs in the advertising links. After processing above, the URLs in the list are all the link addresses of the second pages. But a lot of duplicate links still exist in the list. Later, these duplicate links will be processed. Table 1 shows the URLs list.

Table 1. URLs list

Page number	Page links
2	http://xiamen.baixing.com/pinchesfc/a335134625.html
2	http://xiamen.baixing.com/pinchesfc/a541970104.html
2	http://xiamen.baixing.com/pinchesfc/a551662908.html

Before using the URL list to get the text of carpool data, some measurements should be taken to clean the URL list. All URLs in the collection are filtered to get rid of duplicate link addresses by counting the strings of each URL. After cleaning these link addresses, the link addresses for the second pages is obtained. The URLs of the second pages are ready to be used to get the carpool data from the second web page. As can be seen from the Algorithm flow chart, the find() function

is employed to look for the tag and the results are downloaded to the list. Table 2 shows the results of carpool data.

Table 2. The results of carpool data

车辆情况：	出发地	目的地：	费用：	剩余空位：	出发时间：	途经：
有车	集美灌口	湖滨南路百脑汇	10 元	4 个	周内早 7 点半，晚 5 点半	仙岳路 湖滨中路
有车	杏林湾	杏林大学	8 元	4 个	7:20	324 国道
有车	软件 3 期	杏北广场	面议	4 个	每天早上 7：30	殿前,内林

As the carpool data is non-structural information on the site, in order to facilitate the following statistical analysis work, it is necessary to download some of the data after the processing. It is the same to separate the storage, during which the data are stored in the excel form so that they can be processed in future works.

4.2. Results Analysis

The acquired data will be dealt with such as natural language processing, disambiguation, geo name mapping to map coordinates, statistical analysis and mathematical modeling. And then the geographic information system route planning technology is adopted to get bus routes, starting and ending time. Finally, the starting area and arrival area of the designated bus is determined. After the practice of the test, the theoretical research at the outset is proved to be correct and feasible. In the practical application process, theoretical research is bound to be sufficient and practical.

5. CONCLUSIONS

In the paper, the web crawler program is developed for obtaining the carpool data of large or medium sized cities from the large portal service websites. By analyzing the internet travel data, the paper has found out the right custom route for the bus project. The purpose of designing the web crawler is to provide adequate and reliable data support and utilize resources properly. In the process of designing the web crawler, the gap between theoretical research and practical application is realized. In the process of theoretical research to practical application, a lot of details need to be dealt with. Therefore, the future research requires the combination of theory and practice. The most important point is how to Solve the problem of web page coding. Future work includes the following: based on the acquired large data lanning routes and runnning schedules are established. According to the data calculation results, buses should be arranged to match the passenger travel time and to satisfy the travel location requirements of passengers.

ACKNOWLEDGEMENTS

This work was financially supported by the Scientific Research Funds of Huaqiao University (Grant No. 600005-Z16Y0005). Subsidized Project for Cultivating Postgraduates Innovative Ability in Scientific Research of Huaqiao University (Grant No. 1611314020).

REFERENCES

[1] Bhatia, M. P. S., & Gupta, D. (2008). Discussion on web crawlers of search engine.

[2] Chang, Y. J., Leu, F. Y., Chen, S. C., & Wong, H. L. (2015). Applying Web Crawlers to Develop a Sale Management System for Online Malls. International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (pp.408-413). IEEE.

[3] Bal, S. K., & Geetha, G. (2016). Smart distributed web crawler. International Conference on Information Communication and Embedded Systems (pp.1-5). IEEE.

[4] Chang, Y. J., Leu, F. Y., Chen, S. C., & Wong, H. L. (2015). Applying Web Crawlers to Develop a Sale Management System for Online Malls. International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (pp.408-413). IEEE.

[5] Boldi, P., Codenotti, B., Santini, M., & Vigna, S. (2010).Ubicrawler: a scalable fully distributed web crawler. Software Practice & Experience, 34(8), 711-726.

[6] Liu, F., Ma, F. Y., Ye, Y. M., Li, M. L., & Yu, J. D. (2004). Distributed high-performance web crawler based on peer-to-peer network. Lecture Notes in Computer Science, 3320, 50-53.

[7] Liu, L., Peng, T., & Zuo, W. (2015). Topical web crawling for domain-specific resource discovery enhanced by selectively using link-context. International Arab Journal of Information Technology, 12(2), 196-204.

[8] Mouton, A., & Marteau, P. F. (2009). Exploiting Routing Information Encoded into Backlinks to Improve Topical Crawling. Soft Computing and Pattern Recognition, 2009. SOCPAR '09. International Conference of (pp.659-664). IEEE.

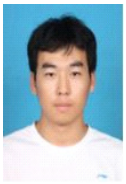
[9] Ghuli, P., & Shettar, R. (2014). A novel approach to implement a shop bot on distributed web crawler. Advance Computing Conference (pp.882-886). IEEE.

[10] Pani, S. K., Mohapatra, D., & Ratha, B. K. (2010). Integration of web mining and web crawler: relevance and state of art. International Journal on Computer Science & Engineering, 2(3), 772-776.

[11] Pesaranghader, A., Mustapha, N., & Pesaranghader, A. (2013). Applying semantic similarity measures to enhance topic-specific web crawling. International Conference on Intelligent Systems Design and Applications (Vol.19, pp.205-212). IEEE.

[12] Chang, Y. J., Leu, F. Y., Chen, S. C., & Wong, H. L. (2015). Applying Web Crawlers to Develop a Sale Management System for Online Malls. International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (pp.408-413). IEEE.

AUTHORS



Junlei Zhao
Postgraduate in College of
Computer Science and Technology, Huaqiao University
Xiamen, China

Bicheng Li
Professor in College of
Computer Science and Technology, Huaqiao University
Xiamen, China