ATLANTIS PRESS

Advances in Engineering Research, volume 166

3rd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2018)

Domain Topic and Hidden Deep Web Data Extracting

Du Liming[a], Yahaya Abdulhamid[b] , Li Gui, Wang Fengying, Dong Jie

Faculty of Information & Control Engineering, Shenyang Jianzhu University, Shenyang, China 110168

[a]duliboy@163.com, [b] Abdulhamidyahaya1@gmail.com

Keywords: Web Data; Data extraction; Data Mining

Abstract. This paper mainly studies the method of extracting web data entities based on domain. Through the analysis of real estate industry websites, a topic-oriented topic extracting model is proposed, and the corresponding search strategy is given. In addition, for the case of depth information, a sorting-based classification extraction algorithm is designed for numerical data. Finally, an experimental example is given to verify the effectiveness of the algorithm.

Introduction

With the development of human society, people's demand for information interaction is increasing, and the Internet has emerged [1]. With the help of the Internet, information can be quickly disseminated, and the types of information are increasing, including documents, pictures, videos, audios, hyperlinks, forms, and so on [1,2,3]. People's demand for information has also grown. The increase in the demand for Web information extraction technology and the in-depth study of the corresponding research work have promoted the development of Web information extraction technology. At present, various types of web information extraction tools and methods have emerged [4,5,6] .

Although most of these tools and systems use web page wrappers to ultimately achieve the acquisition of structured data in a data source, the methodologies used and the areas of research involved are not the same. According to the principle of the method of identifying and locating user's data in web pages, people have roughly classified various web information extraction systems and related technologies. The main types are: ontology based extraction, location based extraction, NLP based extraction, wrapper modeling based extraction, web query based information extraction and so on [7,8,9,10].

However, most of methods applied for extracting web data did not consider the domain requirements, on the other hand, a lot of useful field data are stored in the background database, which belong to hidden deep data, and need continuous query and extraction. Motivated by this, we studies the methods of extracting web data entities based on domains knowledge, and propose a topic-oriented extracting model, for the existence of depth information on the domain webpages, a sorting classification extraction algorithm was designed for numerical data.

Domain Topic Extracting Model and Searching Strategy

Domain Topic Extracting Model. The domain topic extracting model of this article has been improved on the basis of the generic crawler model, and the flowchart of the extracting model used in this paper is shown in the Fig .1.

Compared with the general crawler model, the domain topic model has two more modules: the page topic relevance calculation module and the candidate URL priority calculation module. The page topic relevance calculation module may filter the saved pages according to the relevance of the pages and the topics. If the relevance of the page to the topic is higher than the set threshold, the candidate URL of the page is extracted and input into the candidate URL priority calculation module, and the calculation rules are as follows: If the candidate URL is relatively related to the topic, it is inserted. To the front of the queue, the opposite is inserted into the back of the queue or is discarded.

If the relevance of the page to the topic is lower than the threshold, the webpage is discarded, and the candidate URLs existing in the webpage do not need to be extracted and prioritized. Therefore, it can be seen that these two modules will directly affect the quality of crawled pages.

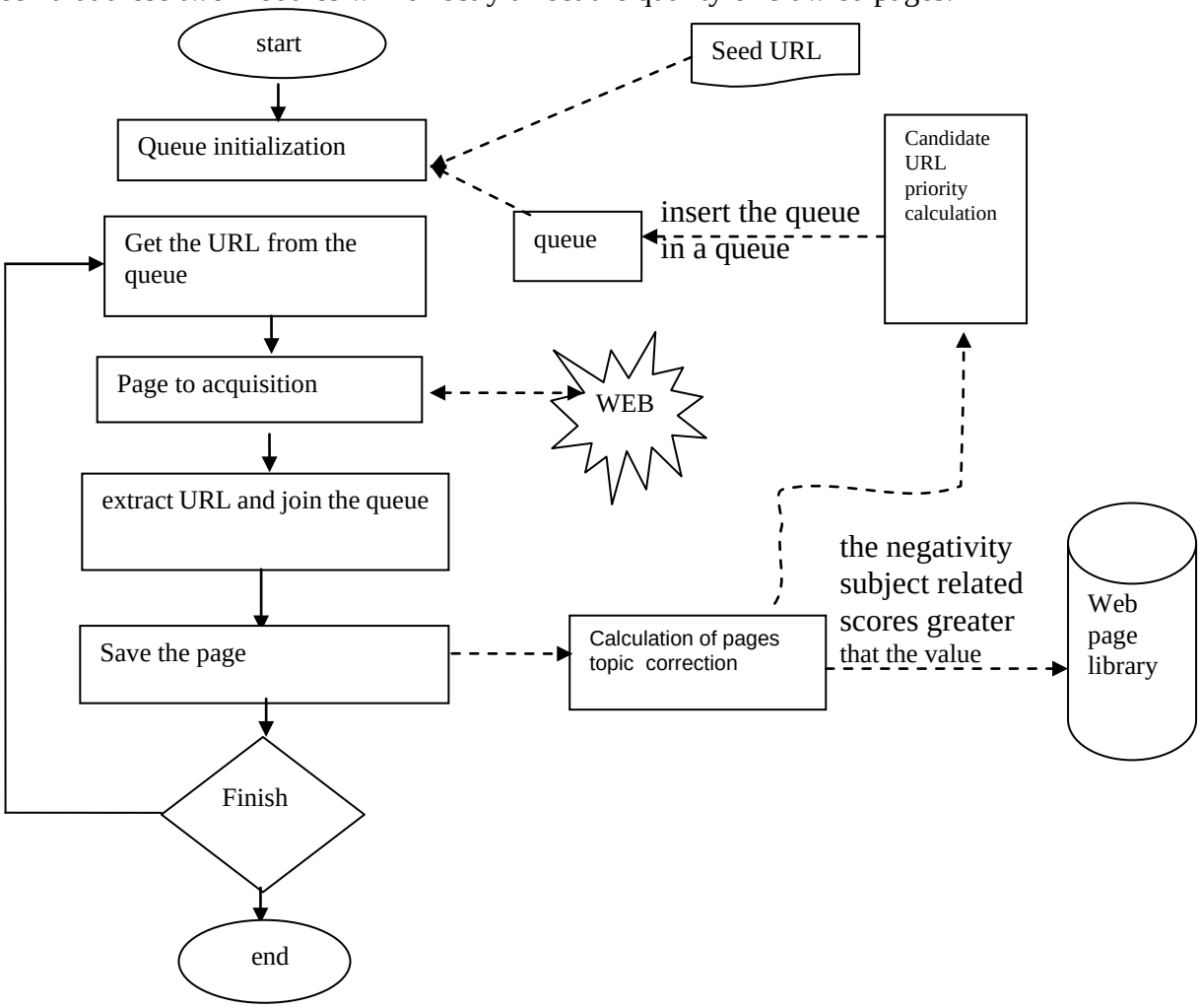

Fig 1 Flow chart of the topic extracting model

The Best Priority Extracting Strategy. Best-first search can be understood as an improvement over BFS. An important principle of the best priority is based on the evaluation letter An important principle of the best priority is to always search for the path that has the least cost based on the evaluation result of the evaluation function. In the search process, the path with the least cost is finally found by continuously giving up the costly path. The topic crawling model using the best priority strategy maintains a URL priority to be crawled during crawling. Rank queues, select high priority URLs from the queue for webpage download, and analyze and calculate these pages Link priority, and then insert the URL priority queue to be crawled according to the level of priority, repeat this process Until the priority queue is empty or the termination condition is reached. The priority of the link depends on the relevance of the web page and the theme, and Web pages with high topic relevance are usually preferentially crawled. Therefore topic crawling models are usually climbed using optimal priority strategies. Take a web page. However, this strategy also has the following features: The URL priority queue space to be crawled is limited, and it is preferred in this queue. The highest-level URL is only temporary, not necessarily the highest global priority, so some of the deep relevance to the topic Layer web pages may be discarded. The Best-First Policy is a simple and efficient best-first search strategy.

Remark 1. Compare with the general crawler model, the domain topic extracting model have the following advantage:

(1) The domain topic extracting model use the optimal priority search strategy better than the common BFS strategy, for the BFS always did not regard the domain requirement.

(2) The general model did not care about the content of net chastity, while the domain topic model extracting the useful data related to the target theme.

(3) The general model will collect web page as many pools possible no matter whether it is used or not, but the domain topic extracting model can avoid downloading unrelated pages.

Hidden Deep Web Data Extracting

Attribute Partitioning Methods. No loss of generality, in this paper we only consider the numerical attributes, and let data space *D* have d-dimensional attributes, the *D* maybe a hidden database in the server, and the server supports queries on database *D* and is implemented for the specific conditions. Specifically, the numerical attribute A_i, its condition is expressed in the form of $A_i \in [x, y]$. Given a query *q*, use *q(D*) express the query result sets, from a formal point of view, according to the size of *q(D),* will lead to two results:

(1) If $|q(D)| \le k$, where $q(D)$ is the entire query resuls, in this case the query *q* has been resolved;

(2) If $|q(D)| > k$, which means that the search result $q(D)$ can only return *k* tupelos, and an overflow flag is returned indicating that the query result overflow.

The *k* value in the above case is the parameter given by the system. If the query result overflows, even if the user repeatedly sends the same query request multiple times, the server will return the same result and cannot obtain the query result. In order to get a complete set of query results, we need to segment the query attributes, here provided two partitioning methods, the first is 2 sections divided, the second is 3 sections divided.

2 Sections Divided. Let $[x_1, x_2]$ is the condition range of q on attribute A_i $(i \in [1, d])$, if the query results $|q(D)| > k$, then take a value $x \in [x_1, x_2]$ (such as $x = \lceil (x_1 + x_2)/2 \rceil$) in attribute A_i, Divide the query q into $q_{left} = q_{(Ai \in [x1,\ x-1])}$ with $q_{right} = q_{(Ai \in [x,\ x2])}$.

3 Sections Divided. Let $[x_1, x_2]$ is the condition range of q on attribute A_i $(i \in [1, d])$, if the query results $|q(D)| > k$ $[x_1, x_2]$, then take a value $x \in [x_1, x_2]$ (such as $x = \lceil (x_1 + x_2)/2 \rceil$) in attribute Ai , which divide the query q into $q_{left} = q_{(Ai \in [x1,\ x-1])}$, $q_{mid} = q_{(Ai \in [x,\ x])}$ and $q_{right} = q_{(Ai \in [x+1,\ x2])}$.

Hidden web data extracting method. It is easy to see that the cost of number partitioning depends on the number of segments of the attribute set. For an attribute $A_i \in [x_1, x_2]$ is random and not evenly distributed, so the number of divisions will not be controlled, and the cost will not be estimated. Suppose the special case that the attributes only have one dimension, then each query result $q(D)$ can be sorted according to the value of attribute A1, so we can take the middle point as the segmentation point, and then based on the query result, the segmentation points and division intervals on Ai is divided into 2 or 3 segments. The algorithm steps are as follows:

Assume the query result for a given query q is $q(D)$, and $q' = q$.

Step 1. First issue a query to the server q', return the result tuple set $R' = q'(D)$. If $|R'| \le k$, the algorithm ends, otherwise the result overflows and step (2) is performed.

Step 2. The tuples in the set R' are sorted according to the value of the attribute A1. Let x be the first $\lceil k/2 \rceil$ tuples, which are intermediate tuples, and let c is the number of R' in $A_1 = x$.

If $c \le k/4$ then we use 2 sections divided method, that means $q'_{left} = q'_{(Ai \le x)}$ with $q'_{right} = q'_{(Ai > x)}$.

If $c > k/4$ then we use 3 segments divided method, that means we get $q'_{left} = q'_{(Ai < x)}$, $q'_{mid} = q'_{(Ai = x)}$ and $q'_{right} = q'_{(Ai > x)}$.

Step 3. Recursively apply the above algorithm and perform loop processing until the query q is resolved. which is let $q'_{left} = q'_{(Ai < x)}$ and return to step (1), let $q'_{right} = q'_{(Ai > x)}$ and return to step (1).

Remark 2. For the general case is that the dimension of attribuite is more than one, then we can calculate the attribuite information entropy, and search for an attributes with maximum entropy gain as sorting attributes.

Experiment

This experiment is based on a real data set (China Land Market Network), where numerical attributes include A1: Signing time, and it is from 1989 to 2016, The land supply results announcement interface contains 1,578,826 tuples, each query page returns 25 tuples. The result of a query returns a maximum of 200 pages, and a total of 5000 tuples are returned. In order to verify the effectiveness of the algorithm, we first crawl the entire hidden database and save it into the local server, then carry out local simulation experiments. The simulation results is as Fig.2.

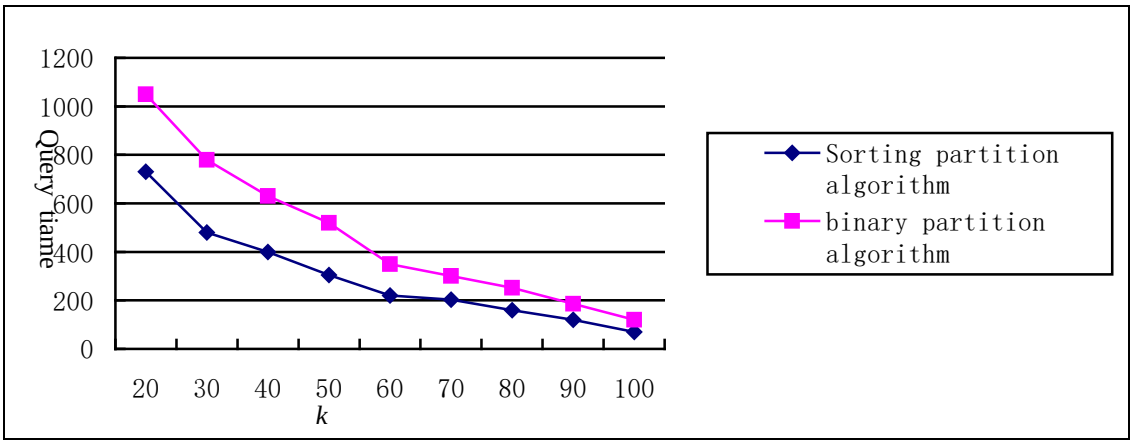

Fig.2. Query cost of numeric algorithms

From Fig.2, we can see that the sorting algorithm is better than the common binary partition algorithm, that is the number of queries is significantly reduced. The cost of the sorting algorithm is inversely proportional to the *k*. This experiment not only confirms the effectiveness of the algorithm, but also proves the efficiency of the algorithm.

Conclusion

The paper provides a domain topic extraction model for extracting the domain web data, the new model has two more modules, which can filter the saved pages according to the relevance of the pages and the topics. Secondly, the paper give a corresponding optimal priority search strategy which is better than the common BFS searching method. Thirdly, for the case of depth information, a sorting-based classification extraction algorithm is designed for numerical data. Finally, an experimental example is given to illustrate the validity of the preseneted method in this paper.

Acknowledgement

This work was supported by the The Ministry of Housing and Urban Rural Development Foundation of China under Grants 2017-K8-038.

References

[1] J. Raskin. "Looking for a Humane Interface: Will Computers Ever Become Easy to Use?", Communications of the ACM, (40:2), 1997, pp. 98-101.

[2] W. Wu, A. Doan, C. Yu, and W. Meng. Modeling and Extracting Deep-Web Query Interfaces. In Advances in Information & Intelligent Systems, pages 65{90, 2009.

[3] W. Su, J. Wang, and F. H. Lochovsky. ODE:Ontology-Assisted Data Extraction. ACM

Transactions on Database Systems, 34(2), 2009.

[4] A. Arasu and H. Garcia-Molina. “Extracting structured data from web pages”. in Proc. 2003 ACM SIGMOD, San Diego, CA, USA, pp. 337–348

[5] Anuradha and A.K Sharma. “Design of Hidden Web Search Engine” . International Journal of Computer Applications (0975 – 8887) Volume 30– No.9, September 2011.

[6] C.-H. Chang and S.-C. Lui. “IEPAD: Information extraction based on pattern discovery,” in Proc. 10th Int. Conf. WWW, Hong Kong, China, 2001, pp.681-688

[7] C. Sherman and G. Price. Hidden Web. “Uncovering Information Sources Search Engines Can’t See”. Cyber-Age book November 2001.

[8] D.W. , Embley et al. "Conceptual-model - based data extraction from multiple-record web page. Data extraction from multiple-record web page". Data and knowledge Engineering31,(199),227-251.

[9] Arasu, A. and Garcia-Molina, H. "Extracting structured data from Web pages. Proceedings of the ACM SIGMOD International Conference on Management of Data". San Diego, California, pp. 337-348, 2003.

[10] Ribeiro-Neto, B., A., Laender, A., H., F. and DA Silva, A., S. "Extracting semi-structured data through examples". Proceedings of the Eighth ACM International Conference on Information and Knowledge Management (CIKM), Kansas City, Missouri, pp. 94-101, 1999.