International Journal of Grid Distribution Computing
Vol.8, No.2 (2015), pp.49-60
http://dx.doi.org/10.14257/ijgdc.2015.8.2.06

A Focused Crawler URL Analysis Algorithm based on Semantic Content and Link Clustering in Cloud Environment

Mingming Li[1], Chunlin Li[1], Chao Wu[1] and Youlong Luo[2]

[1]*School of Computer Science, Wuhan University of Technology, Wuhan, P.R. China*
[2]*Management School, Wuhan University of Technology, Wuhan, P.R. China*
610087798@qq.com, chunlin74@aliyun.com, 124528721@qq.com, 124528721@qq.com

Abstract

Currently, the efficiency of the existing focused crawlers is not high because of their unsatisfactory precision. In this article, we analyze the URL analysis methods of the existing focused crawlers, and propose a URL analysis algorithm based on the semantic content and link clustering in cloud environment. In this algorithm, the download URLs are clustered with the philosophy of clustering on the basis of VSM to improve the precision of the focused crawler according to the correlation between download URLs and new URLs. The algorithm is evaluated on Heritrix3.10 compared with Best First Search algorithm and Shark Search algorithm. The experiment results demonstrate that the algorithm proposed can collect web pages related to the given topic accurately and effectively.Moreover, the algorithm has a good ability of learning which proves the possibility of this algorithm.

Keywords: *Focused Crawler, URL analysis algorithm, DBSCAN, Map/Reduce, Semantic Content*

1. Introduction

As the Internet develops sharply and vast amounts of Web resources appear on the Internet in recent years, vertical search engines with focused crawler as the core component become more and more popular among users for its pertinence and accuracy. Vertical search engines provide Web query based on semantic content with domain knowledge in a given industry or theme [1]. Focused crawlers make more satisfactory results than general search engines on a small scale of data to meet the needs of specific users, because general search engines have many limitation such as low coverage ratio [2]. Focused crawlers only download webpages in a certain industry or a certain theme, so focused crawlers are more targeted to satisfy users' special requirements. They get most of data in a given area while consuming few server resources for the reasons that professional focused crawlers work in a small area, the results they return are more pertinence and they index a small percentage of the world-wide-web data only.

The core problem of focused crawler is choosing algorithm to determine the correlation from pages downloaded and URLs from the pages to the given topic. The existing algorithms are divided into three kinds: analysis based on link structure, analysis based on web content and analysis based on semantic content. The first method may lead to topic drift because it considers URLS only while ignoring the relation between URLs and the given topic. The second method consumes too much computing resource as it analyses every page. The last one ignores the correlation among URLS in contrast with the analysis based on link structure. The download speed of focused crawlers is also an factor that restricts their efficiency, compared to general crawlers.

ISSN: 2005-4262 IJGDC

Based on semantic content analysis, this paper uses DBSCAN algorithm with Map/Reduce model to cluster download URLs and thus analyze web links. This algorithm proposed in this paper takes not only the relation between URLS and topics into account, but the correlation between download URLs and new URLs; To improve the download speed of crawlers, the algorithm is realized with Map/Reduce model;

The remaining sections of this paper is organized as follows: The second section introduces existing web analytics strategies of focused crawlers; Section 3 presents a new web-based content and links clustering analysis algorithm compared to the existing method; Section 4 illustrates the experiment results which demonstrates the superiority of our methods; Section 5 summarizes the study and points out the direction of future work.

2. Related Work

In the domain of vertical search engines, themed websites are generally created by manual processing based on the results returned by focused crawlers. Many analysis methods have appeared, and the following is the analysis and summary.

Literature [3] proposed a filter method based on DOM Tree according to the characteristics of the oil industry and discussed the strategy, the design and implementation of themed search engine. Literature [4] used the vector space model to calculate the similarity of pages with the improved Shark-Search algorithm and determined the list of URLs to be downloaded. It improved the precision of pages by collection URLs related with multi-thread method. An improved HITS algorithm was raised in literature [5] to calculate the value of URL by assigning URLs with the similarity between semantic content and topic. Liu proposed a semantic similarity vector space model (SSVSM), and determined the web page relevance by quantifying the web text and link anchor text respectively [6]. Kokai built up a focused crawler with SVM and reinforcement of learning method [7]. Pant constructed a multi-threaded crawler with SVM classifiers [8]; Chakrabarti classified the pages downloaded by constructing a classifier to decide whether the page is related [9]. Li used decision tree to forecast new URLs by evaluating the anchor text [10]. Some of the researches above estimate URLs based on semantic content only while ignoring the information from URLs and some estimated URLs on the contrary.

The straightforward idea is to utilize features of URLs, as the texts in the URLs have meanings which can represent the categories of those webpages. Many works have studied the practicability of fast and pure URL-based classification methods [11-13]. In these methods URLs are commonly mapped to feature vectors, and then a few traditional classification algorithms are adopted to classify these vectors. Blanco [13] proposed a highly efficient algorithm which not only applied string similarity, but also utilized the features of scripts. By distinguishing URL segments between data terms and script terms, the webpages generated by the same script are identified as the same classification. Inspired by this method, it can be speculated that if some webpages of a single website are relevant to the same topic, their URLs have a common feature.

Several studies focused on crawlers based on content-based evaluation and link structure. The former algorithms with traditional information in vector space, used the domain knowledge message from pages to guide the search, such as Best-First Search, Fish-Search and Shark-Search, *etc.*, [14].The classifier to determine the relevance of a Web page in Fish-Search algorithm is called Binary classifiers [15], compared to which the Shark Fish algorithm has higher accuracy [16]. The latter is mainly based on analysis of web links to dig subject information. The main strategy is to use the HITS algorithm and PageRank algorithm to evaluate the relevance of URLs to the topic [17]. Maimunah divided web link analysis into four groups and

improves the recall ratio by designing the crawlers for every structure of links [18]. Such algorithm considered the structure of URLs and page content, but they paid no attention to the relevance of download URLs and URLs to be estimated; therefore this factor will be introduced in this paper.

3. The Algorithm based on Semantic Content and Link Clustering

3.1 Vector Space Model

Vector Space Model (VSM) measures the similarity of topic relevance with data and set in vector space [19]. The model is a simple but efficient file representation model in which every text file is mapped to a vector consisting of standard orthogonal entries. All documents can be represented by vector $\{w_1, w_2, ..., w_i, ... w_n\}$, there into w_i is the weight of entry t_i . The representation and matching of text are converted to those of vectors. Suppose the vector of target document $q = \{X_1, X_2, ..., X_i, ..., X_n\}$ and the vector of new document $p = \{Y_1, Y_2, ..., Y_i, ..., Y_n\}$, then the similarity of these documents equals cosine of these vectors. The similarity is calculated as follows:

$$sim(q, p) = \frac{\sum_{i=1}^{n} X_i * Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2} \sqrt{\sum_{i=1}^{n} Y_i^2}} \quad (1)$$

The advantage of vector space model is that it simplifies the computational complexity of the page relevance, and makes calculating converse to operations based vector space model. Therefore vector space model will be widely used in this algorithm to calculate the similarity of URLs.

3.2. Judgments of Page Relativity

When focused crawlers download related web pages, the URLs extracted needs to get through topic relativity judgments, and that is the biggest difference between vertical search engine and general search engine. How to quickly filter out the web pages less relevant to the topic is the key to deciding search strategy for focused crawlers. Thus after extracting page text messages, the topic relativity judgment of pages should be made so as to filter out irrelevant web pages to the topic and retain high relevant web pages.

At present, most studies use vector space model as a method to judge the topic relativity of pages. This method maps page documents to feature vectors in VSM. Calculate the inner product of collected pages' vector and topic's vector to evaluate collected pages according to the matching and measurement of VSM model: the larger the inner product is, the higher the topic relativity of collection pages is [20], and the calculation formula is shown as follows.

$$Sim(D_i, D_j) = \frac{\sum_{k-1}^{N} w_{ik} * w_{jk}}{\sqrt{\sum_{k-1}^{N} w_{ik}^2} * \sqrt{\sum_{k-1}^{N} w_{jk}^2}} \quad (2)$$

Thereinto, D_i is the theme, D_j is the page to be estimated, they are represented by vector $(d_{i1}, d_{i2}, \cdots, d_{in})$ and vector $(d_{j1}, d_{j2}, \cdots, d_{jn})$ respectively. Among them, w_{ik}, w_{jk} respectively correspond to the weight of the k -th in D_i, D_j pages, w_{ik}, w_{jk} are

quantitatively processed by TF-IDF method. Compare $sim(D_i, D_j)$ ' s value with the threshold value. If the former is greater than or equal to the threshold value the page is relevant to the topic and keep it in the database, otherwise discard the page because it is estimated to be irrelevant.

3.3. Clustering Downloaded URLs with DBSCAN Algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm based on density. Its basic idea is that for each object clusters, the number of objects contained within the neighborhood ϵ must be equal to or greater than a constant value (MinPts), that is to say the density of its neighborhood ϵ must be not less than a threshold value. The algorithm searches for all the objects density reachable in a cluster iteratively [21].

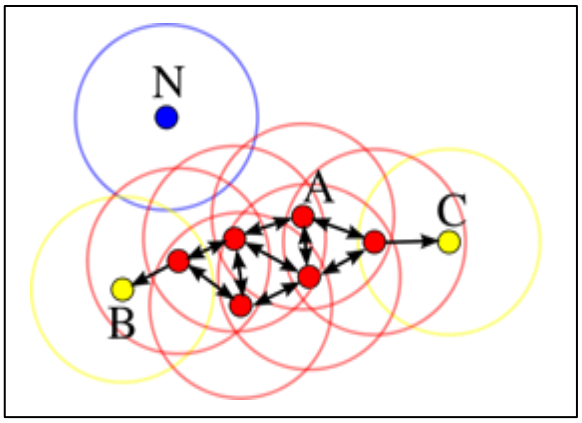

Figure 1. An Example of DBSCAN Algorithm

(Points at A are core points. Points B and C are density-reachable from A and thus density-connected and belong to the same cluster. Point N is a noise point that is neither a core point nor density-reachable. (MinPts=3 or MinPts=4))

As most web pages are generated dynamically, the URLs in the same portal website have many similarities. Now take the military project of China Net as an example:

The address of home page is: *http://military.china.com/zh_cn/*

Then select three sub links about news arbitrarily:

http://military.china.com/important/11132797/20140617/18566076.html
http://military.china.com/important/11132797/20140617/18566112.html
http://military.china.com/important/11132797/20140615/18562742.html

Two points could be found: their structure and content links are very similar; besides, the text of URLs contains the word "military". Consequently, a strategy analyzing the relationship between the texts of URLs and keywords of given topic is proposed to establish the correlations among URLs by building a Library of download URLs clustered. The URL is split by "/" and "." to be a vector $P = (p_1, p_2, \cdots, p_i, \cdots p_n)$.

According to the similarity of URL text structure and content from the same web portals , these pages should be divided together with DBSCAN clustering algorithm to create a link cluster $R = (r_1, r_2, \cdots, r_i, \cdots r_n)$ in which $U = (u_1, u_2, \cdots, u_i, \cdots, u_n)$ represents the number of URLs. The following formula is used to calculate "the blood relationship" between new URLs and download URLs.

$$Sim(url, urls) = \sum_{i=1}^{n} sim(r_i, P) * u_i \qquad (3)$$

On the other hand, formula (4) can help calculate the similarity value between link vector and target topic Q for the sake of the similarity between new URL's text and keywords in the given topic.

$$Sim(url\_content) = \sum_{i=1}^{n} sim(p_i, Q) \quad (4)$$

Finally, get F, the similarity value of new URL, with formula (3) and formula (4):

$$F(P) = \omega * Sim(url, urls) + (1-\omega) * Sim(url\_content) \quad (5)$$

where $0 \le \omega \le 1$ is a weight

The Figure below shows the process after clustering the download URLs and analyzing the new URLs. As URL A has high similarity with cluster A and cluster A has a large number of pages (1000), the value F of URL A is the biggest of all. To URL C, by contrast, has no cluster that is similar, its value is minimum. So the values of them are sorted as below:

$$F(A) > F(B) > F(C)$$

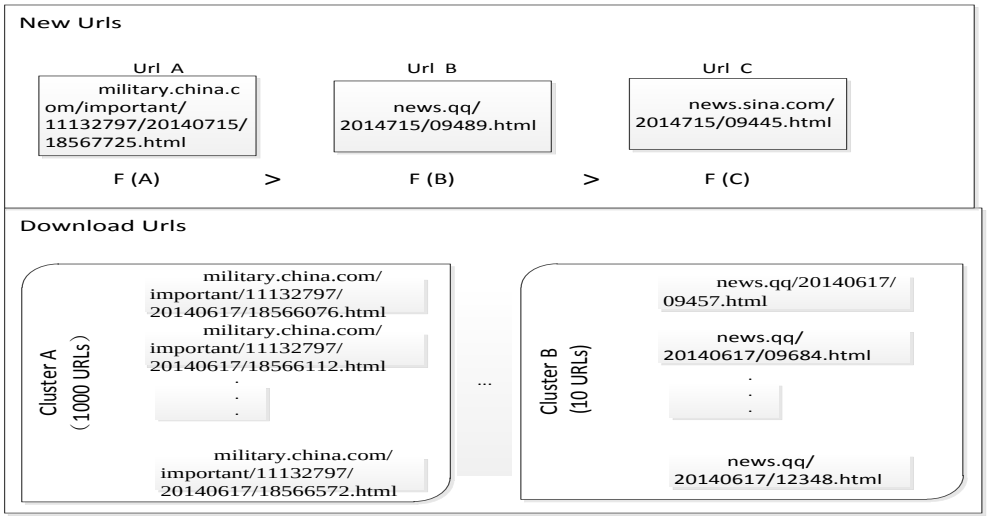

Figure 2. The Similarity Estimating of URLs Clustered

3.4. Algorithm based on Semantic Content and Link Clustering

The algorithm based on the analysis strategy of semantic content introduces new URL similarity mentioned in last section, and remark the URL to choose the next URL to download for focused crawlers.

When a Web page is downloaded, new URLs will be parsed out. The number of useful URLs from the page depends on whether the page is correlated to the given theme and the number of URLs it contains. The key of the algorithm is to maintain a URL priority queue dynamically with the seed links and topic keywords. When a webpage is grabbed, pages of URLs contained are called "child page". Similarity measure method is introduced to calculate the weight of new URL with not only that of parent node, but also the anchor text.

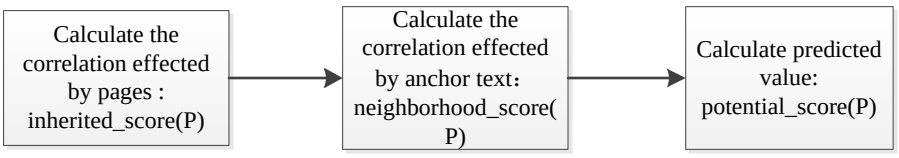

Figure 3. The Flow Chart of Analysis based on Semantic Content

The general idea of this algorithm is shown in Figure 3 and Figure 4: firstly get the related predicted correlation value of new URL based on the analysis of semantic content, which is similar to the way of Shark-Search algorithm; then use DBSCAN algorithm to cluster downloaded URLs regularly or quantificationally, and calculate the new URL's similarity according to clustering results. This algorithm combines the analysis based on semantic content with the similarity between download URLs clustered and the target topic to remark the URL with the following formula:

$$final\_score(P) = \varphi * potential(P) + (1-\varphi) * F(P) \quad (6)$$

where $0 \le \varphi \le 1$ is a weight

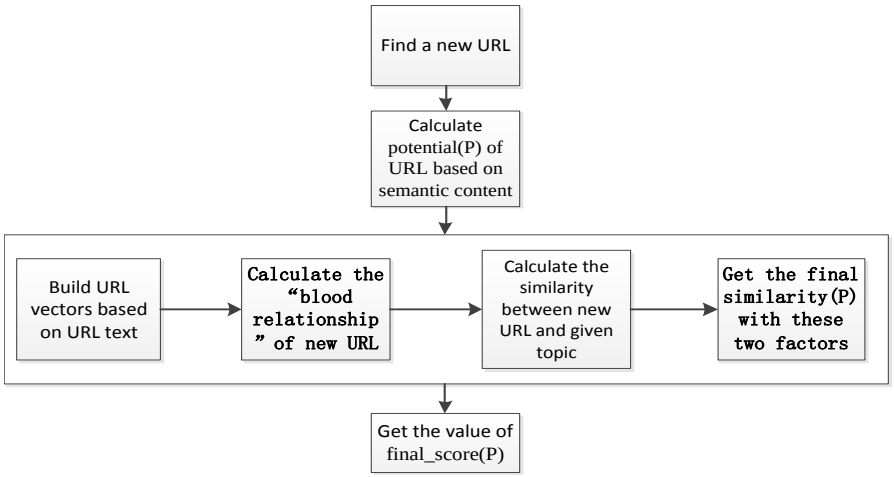

Figure 4. Model of Algorithm Proposed

4. Experiment Result and Analysis

In this section, the optimal values of the parameters ω、φ in the algorithm are confirmed with experiment. By comparing the algorithm with existing Best-First Search algorithm, Shark-Search algorithm and analyzing the result, the superiority of the algorithm proposed is demonstrated.

The experimental platform takes Heritrix3.10 as its framework with hadoop-0.20.2 as its environment, and "Military" as its topic. Select 10 websites related to the topic while 15 websites not related, and set them to be the experimental test set. The correlation of pages downloaded is estimated by extraction page text and computation theme relevancy after the secondary development of Extractor components. Extend FrontierSchedular component to realize the URL search strategy and continuously update the URL queue until the end of the crawling process. Set the number of seed URLs which include those related to military theme 25 and the number of test pages 4000. Cluster the download URLs when another 100 new URLs are downloaded. Count the number of pages related to given topic while the number of pages downloaded varies from 500 to 4000, then calculate the recall ration, precision ratio according to the experimental data.

To determine the best value of parameters ω and φ in formula (4) and formula (5), control variants method is taken in experiment: Respectively observe the experimental results when one parameter changes from 0 to 1 under the condition that another parameter reaches its limit value 0 and 1.

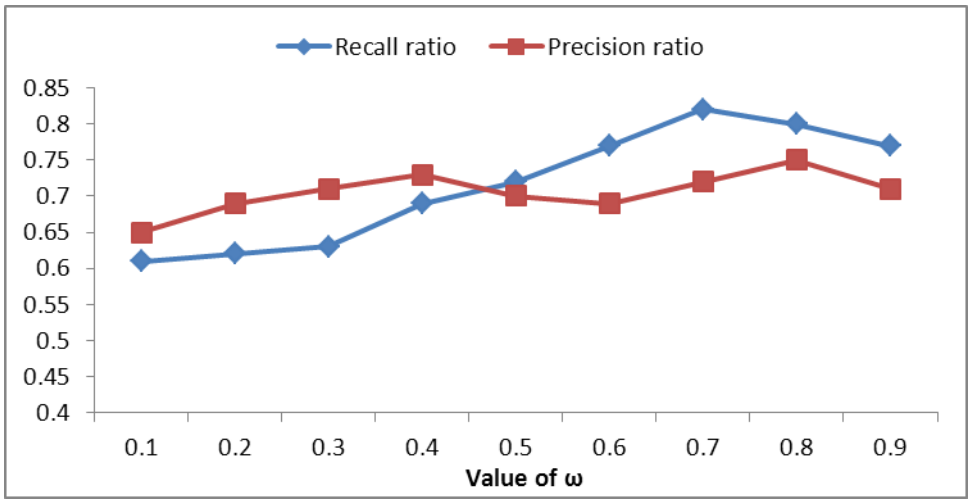

Figure 5. The Recall Ratio and Precision Ratio of Different ω

Firstly, set the value of φ 0 and the pages' number 4000, then the algorithm proposed is only affected by *F(p)*. Observe the effects of parameter ω on the recall ratio and precision ratio which are shown in Figure 5: when ω equals to 0.7, the recall ratio reaches its maximum 0.82 while the precision ratio which always fluctuates up and down near 0.7 is not affected. Therefore consider 0.7 to be the best value in this algorithm. The formula is as follows:

$$F(P) = 0.7 * Sim(url, urls) + 0.3 * Sim(url\_content) \quad (7)$$

The value of ω indicates that the clusters of URLs affect more to the similarity when compared with URL's text; but the latter is also an important factor that can't be ignored. They complement each other.

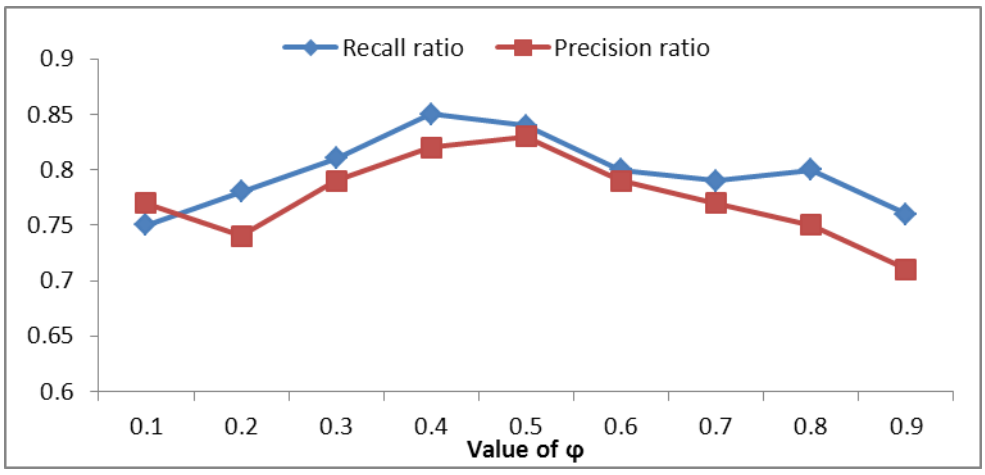

Figure 6. The Recall Ratio and Precision Ratio of Different ϕ

Secondly, determine the value of parameter φ with the same method: Set the value φ 0.7, the number of pages 4000, observe the influence of parameter φ on recall ratio and precision ratio in the algorithm, the result is shown in Figure 6. As the result shows: when parameter φ values 0.4 and 0.5, the recall ratio and the precision ratio reach their maximum value, 0.85 and 0.83. Considering of these two factors, 0.45 is determined to be the value of φ and the formula is as follows:

$$final\_score(P) = 0.45 * potential(P) + 0.55 * F(P) \quad (8)$$

It is not difficult to find the importance of analysis based on semantic content in this algorithm when contrasting figure 5 and figure 6. The recall ratio and precision ratio increase distinctly with semantic content added into the algorithm. The value of φ is 0.45, the weight of semantic content balances that of links' similarity which means they are all important to URLs analyzing and either of them could be ignored.

At last, ω=0.7 and φ=0.45 have been determined to be the best value of parameter ω, φ. To demonstrate the superiority of our algorithm, Best First Search and Shark Search are compared on the recall ratio and precision ratio.

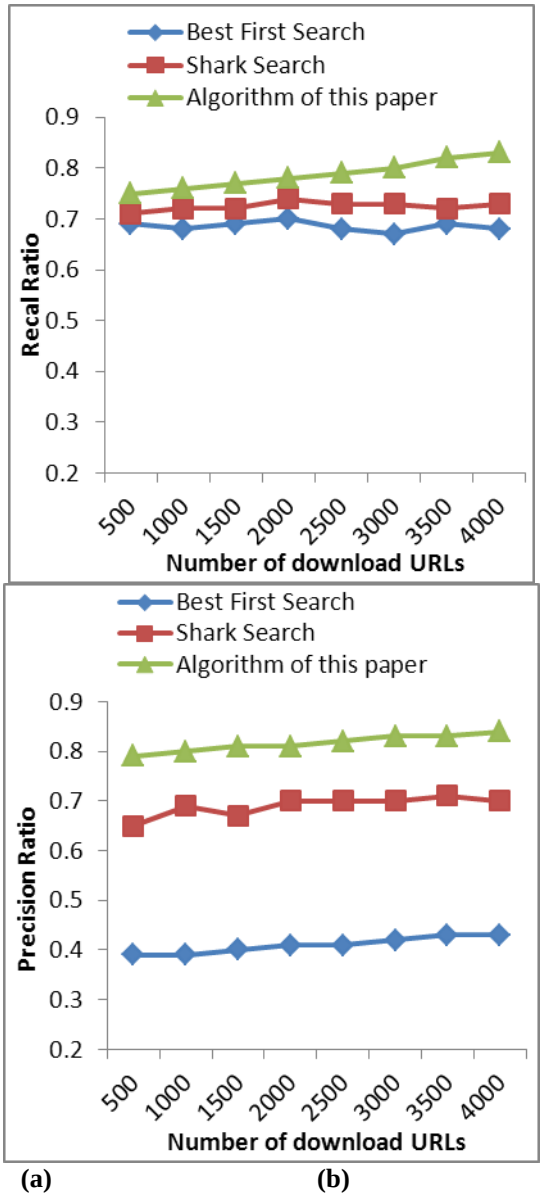

Figure 7. The Contrast Among Three Algorithms

As shown in Figure 7, algorithm mentioned in the text based on the analysis of semantic content and URLs clustering has great superiority compared with Best-First Search algorithm and Shark - Search algorithm when $\omega=0.7$, $\phi=0.45$. The recall ratio and precision ratio in this algorithm are higher than those in the latter two methods. That is because Best-First Search algorithm and Shark-Search algorithm only consider the relationship between URLs and topic, but they ignore the correlation among URLs,

leading to the low efficiency of crawlers. Thus it can be seen that comprehensively considering the relevance between URLs and topics, the relationship between downloaded URLs and new URLs is of certain help to improve the performance and efficiency. The algorithm clusters downloaded URLs and estimates new URLs with the similarity of URLs' texts; this process acts as a URL classifier in crawlers, and improve the sensitivity to know new URLs' websites and thus increase the capacity of estimating URLs; what's more, this algorithm has the ability of self-learning: along with the quantity of downloaded URLs increasing, the recall ratio and precision ratio improve for the reason that the reference value to estimate URL grows continually with referable URL clusters increasing constantly.

Table 1. Comparison of the Proposed Method with Shark-search and Best-First

Algorithm	Number of pages download	Time(seconds)	Pages download per second	Recall	Precision
Best-First	1000	19.2	52.1	0.68	0.39
Best-First	4000	76.9	52.0	0.68	0.43
Shark-search	1000	20.0	50.0	0.72	0.69
Shark-search	4000	79.5	50.3	0.73	0.70
Algorithm in this paper	1000	20.4	49.1	0.76	0.80
Algorithm in this paper	2500	51.2	48.8	0.79	0.82
Algorithm in this paper	4000	82.9	48.2	0.83	0.84

On the other hand, as the time complexity of DBSCAN is O (n^2), the consuming of the algorithm proposed in this paper is a question to be considered when the number of downloaded pages is huge. The algorithm clusters download URLs by default when another 100 new URLs are downloaded. Now we set 4 threads work at the same time in the crawler, and compare these three algorithms. The result is shown in table 1.

According to the results in the table 1, the downloading speed of Best-First algorithm is the highest, reaching 52.1 and 52.0 pages per second. But its Recall ratio and Precision ratio can't compare to those of Shark-Search algorithm and the algorithm proposed by this paper. Contrasting to Shark-Search, the algorithm in this paper has a litter lower speed but much higher Recall ratio and Precision ratio. So the downloading speed of this algorithm is acceptable with its high efficiency. The algorithm achieves more satisfactory results with certain additional resources consumption.

5. Conclusion and Future Works

In this paper, we conclude that there are mainly three contributions:

(1) According to the relations between download URLs and new URLs, a concept of mapping every URL to a vector is proposed. As URLs related to the same topic are similar, it can help calculate the similarity of URL by utilizing VSM (Vector Space Model).

(2) Based on the philosophy of clustering, DBSCAB algorithm is applied to cluster downloads URLs into many clusters each of which is composed of similar webpages. The clusters can help estimate whether the new URL is related to the topic or not.

(3) According to above concepts and algorithms, a new algorithm of URL selecting which combine semantic content and link clustering is proposed. The best value of parameters in the algorithm is determined by experiments. The algorithm makes full use of the relationship between new URLs and URLs downloaded, and improves the accuracy of the focused crawlers.

As experiment results show, the algorithm has two advantages compared to others of the same kind: the pages are downloaded accurately , effectively, and the algorithm has a good ability of learning which proves the possibility of the algorithm. And the cloud environment provides secure expandable storage and Map/Reduce model.

But in this algorithm, the calculation of topic similarity based on links and text information in pages should be more accurate. Thus a new method mapping keywords to the level of semantic concept should be taken to analyze the topic relevance of page text on words' semantic level. And it is necessary to reduce the complexity of the clustering algorithm and the frequency of clustering. The above is the direction of future works.

Acknowledgements

The work was supported by the National Natural Science Foundation (NSF) under grants (No.61472294, No.61171075), Program for the High-end Talents of Hubei Province, Key Natural Science Foundation of Hubei Province (No. 2014CFA050), the Fundamental Research Funds for the Central Universities(WUT:2014-Ⅶ-027) , and Open Fund of the State Key Laboratory of Software Development Environment (SKLSDE). Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

References

[1] B. B. Cambazoglu, E. Varol and E. Kayaaslan, "Query forwarding in geographically distributed search engines", In SIGIR, **(2010)**, pp. 90-97.
[2] V. Gil-Costa, A. Inostrosa-Psijas and M. Marin, "Service Deployment Algorithms for Vertical Search Engines", Parallel, Distributed and Network-Based Processing (PDP),2013 21st Euro-micro International Conference on IEEE, **(2013)**, pp. 140-147.
[3] L. Dangwei and P. Wentao, "Designing and realizing the filter in vertical search engine", Computer Applications and Software, vol. 26, no. 12, **(2009)**, pp. 148-151.
[4] Z. Bo and C. Wandong, "The Research and Realizing of Focused Crawler Technique", Microelectronics & Computer, vol. 26, no. 5, **(2009)**, pp. 52-55.
[5] A. Patel, "An adaptive updating topic specific web search system using T-Graph", Journal of Computer Science, vol. 26, no. 4, **(2010)**, pp. 450-456.
[6] L. W. Duy, "An Improved Topic-Specific Crawling Approach Based on Semantic Similarity Vector Space Model", Journal of Computational Information Systems, vol. 8, no. 20, **(2012)**, pp. 8605-8612.
[7] A. Lorincz, I. Kokai and A. Meretei, "Intelligent High-Performance Crawlers Used to Reveal Topic-Specific Structure of the WWW", International Journal of Foundations of Computer Science, vol. 13, no. 4, **(2002)**, pp. 477-495.
[8] G. Pant and P. Srinivasan, "Link Contexts in Classifier-Guided Topical Crawlers", Knowledge and Data Engineering, IEEE Transactions, vol. 18, no. 1, **(2006)**, pp. 107-122.
[9] R. Campos, O. Rojas, M. Marin, *et al.*, "Distributed Ontology-Driven Focused Crawling", Parallel Distributed and Network-Based Processing (PDP), 2013 21st Euro-micro International Conference on IEEE, **(2013)**, pp. 108-115.
[10] J. Li, K. Furuse and K. Yamaguchi, "Focused Crawling by Exploiting Anchor Text Using Decision Tree", Special Interest Tracks and Posters of the 14th International Conference on World Wide Web ACM, **(2005)**, pp. 1190-1191.

[11] M. Kumar and R. Vig, "Multilingual Context Ontology Rule Enhanced Focused Web Crawler", University Institute of Engineering and Technology, Panjab University, India, Journal of Advances in Information Technology, **(2010)**.
[12] D. Hati and A. Kuma, "UDBFC: An Effective Focused Crawling Approach Based On URL Distance Calculation", Computer Science and Information Technology, **(2010)**.
[13] L. Blanco, I. Dalvi and S. Machanavajjhala, "Highly Efficient Algorithms for Structural Clustering of Large Websites", In: WWW, **(2011)**.
[14] Y. Uemura, T. Itokawa, T. Kitasuka, *et al.*, "Where to Crawl Next for Focused Crawlers", In: Springer, Heidelberg, **(2010)**.
[15] R. Setchi, I. Jordanov, R. J. Howlett and L. C. Jain, (eds.) KES, vol. 4, no. 6279, **(2010)**, pp. 220–229.
[16] E. Feuerstein, V. G. Costa, M. Mizrahr, *et al.*, "Performance Evaluation of Improved Web Search Algorithm", In VECPAR, **(2012)** pp. 236-250.
[17] L. Peng and T. Wen-Da, "An Improved Shark-Search Algorithm", Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on IEEE, **(2010)**, pp. 512-516.
[18] F. Bussche, "Not so creepy crawler: easy crawler generation with standard XML queries", Proceedings of the 19th International Conference on WWW, Raleigh, North Carolina, USA, **(2010)**, pp. 1305-1308.
[19] S. Maimunah, H. S. Sastramihardja, D. H. Widyantoro, *et al.*, "CT-FC: More Comprehensive Traversal Focused Crawler", TELKOMNIKA Indonesian Journal of Electrical Engineering, vol. 10, no. 1, **(2012)**, pp. 189-198.
[20] A. Pal, D. S. Tomar and S. C. Shrivastava, "Effective Focused Crawling Based on Content and Link Structure Analysis", arXiv Preprint arXiv, vol. 0906, no. 5034, **(2009).**
[21] S. W. Zhu, J. J. Wu, *et al.*, Scaling up Top-K cosine similarity search, Data& Knowledge Engineering, vol. 10, no. 1, **(2011)**, pp. 60–83.
[22] M. N. Gaonkar and K. Sawant, "Auto Eps DBSCAN: DBSCAN with Eps Automatic for Large Dataset", Published in International Journal on Advanced Computer Theory and Engineering (IJACTE), ISSN (Print), **(2013)**, vol. 2, pp. 2319-2526.

Authors

Mingming Li, is a M.S. student in the Department of ComputerScience at Wuhan University of Technology He received his B.S. degree in Software Engineering from Guilin University of Electronic Scienceand Technology in 2012. His research interests are in cloud computing.

Chunlin Li, is a Professor of Computer Science at Wuhan University of Technology. She received her M.S. in Computer Science from Wuhan Transportation University in 2000 and her Ph.D. in Computer Software and Theory from Huazhong University of Science and Technology in 2003. Her research interests include cloud computing and distributed computing.

Chao Wu, is a M.S. student in the Department of ComputerScience at Wuhan University of Technology. He received his B.S. degree in Software Engineering from Huazhong University of Science and Technology Wuchang Branch in 2012. His research interests are in cloud computing.

Youlong Luo, is a vice Professor of Management at Wuhan University of Technology. He received his M.S. in Telecommunication and System from Wuhan University of Technology in 2003 and his Ph.D. in Finance from Wuhan University of Technology in 2012. His research interests include cloud computing and electronic commerce.