TELKOMNIKA, Vol.12, No.4, December 2014, pp. 1105~1112
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v12i4.981

A Model of Vertical Crawler using Hidden Markov Chain

Ye Hu, Jun Tu*, Wangyu Tong
School of Computer Science and Technology, Hubei University of Technology, No. 1 NanHu Street, Hongsan 430068,Wuhan, China
* Corresponding author, e-mail: tujun_hut@163.com

Abstract

The large size and the dynamic nature of the Web make it necessary to continually maintain Web based information retrieval systems. In order to get more objects by visiting few irrelevant web pages, the web crawler usually takes the heuristic searching strategy that ranks urls by their importance and preferentially visits the more important web pages. While some systems rely on crawlers that exhaustively crawl the Web, others incorporate "focus" within their crawlers to harvest application or topic-specific collections. However, very limited work has addressed the urgent issue of how to measure the crawled page's importance. In order to deal with this issue, this paper has presented a new model taking advantages of the Hidden Markov Model (HMM) to enhance the crawler performance. In this new model the improved HMM learning ability is employed to solve the problem of the theme of the crawler drift and thus improve the data acquisition accuracy. Numerical simulation tests have been carried out to verify the performance of the proposed approach. The analysis results have shown that the new model was very efficient and was superior to the existing Bayesian probabilistic model, Naïve Bayes model, and K-Nearest Neighbor Approach. Thus, the proposed improved HMM based approach can be used in practice.

Keywords: *hidden markov model, crawler, uniform resource locator*

1. Introduction

Due to the explosive growth of the web pages, besides to hope the search engine that can provide more and more appropriate information, people have the requirement of taking centralized query on given topic. The dynamism of the Web, crawling forms the backbone of applications that facilitate Web information retrieval. While the typical use of crawlers has been for creating and maintaining indexes for general-purpose search engines, diverse usage of topical crawlers is emerging both for client and server-based applications. Topical crawlers are becoming important tools to support applications such as dynamic Web portals, online searching, and so on.

A large number of algorithms have been proposed for building crawlers. The difference is in the heuristics they use to score the unvisited URLs, with some algorithms adapting and tuning their parameters before or during the crawl [1].

In [2] Chakrabarti et al. described a focused crawler, that searches the web to find relevant pages on a given topic. The crawler utilizes a classifier to determine relevancy of a page and a distiller to evaluate page links. A comparison of learning schemas employed by focused crawlers can be found in [3]. In order to determine whether a web page matches with a predefined topic, classification algorithms are employed. Some of the classification algorithms are Bayesian probabilistic model [4], Naïve Bayes model [5], K-Nearest Neighbor Approach [6] etc. In [7], Dixit described the mechanism called migrating crawler to reduce load on the network by sending the migrants to the web server itself for taking the advantage of local downloading and filtering before sending the documents to the Search engine repository.

Ranking search results is a fundamental problem in information retrieval. To present the documents in an ordered manner, Page Ranking methods are applied, which can arrange the documents in order of their relevance, importance and content score. Some of the common page ranking algorithms are PageRank Algorithm [8], Weighted Page Rank Algorithm [9] and Hyperlinked Induced Topic search Algorithm [10]. The PageRank algorithm provides a global ranking of Web pages based on their importance estimated from hyperlinks [8]. For instance, a link from page "A" to page "B" is considered as if page "A" is voting for the importance of page "B". So, with increase in number of links to page "B", its importance also increases.

Received September 4, 2014; Revised November 13, 2014; Accepted November 29, 2014

The PageRank algorithm attempts to provide a global estimate of Web page importance. However, the importance of Web pages is subjective for different users and thus can be better determined if the PageRank algorithm takes into consideration user preferences. The importance of a page may depend on different interests and knowledge of different people therefore a global rank may not provide the actual the importance of that page for a given individual user.

With increasing popularity of search engines, implicit feedback, i.e., the actions users take when interacting with the search engine, can be used to improve the rankings. Implicit relevance measures have been studied by several research groups. An overview of implicit measures is compiled in Kelly and Teevan [11]. However, the findings of the research work, were not applied to improve the ranking of web search results in realistic settings. What we need is a measure of the crawled page's importance, and then a method to summarize performance across a set of crawled pages. A number of topical crawling algorithms have been proposed in the literature. Often the evaluation of these crawlers is done by comparing a few crawlers on a limited number of queries/tasks without considerations of statistical significance. For example, the existing ranking algorithms mainly estimate the url's importance by web pages' relevancy to the topic or their authorities. There are several common algorithms for evaluating authorities, such as pagerank, leinberg, hits and salsa [12]. These algorithms can exactly evaluate the web page's authority, but they hardly consider topical information, resulting in a problem of topic-drift that means although the web page with high authority score certainly has high universal authority, it not always has high authority on given topic too [13]-[17].

In general, it is important to compare topical crawlers over a large number of topics and tasks. This will allow us to ascertain the statistical significance of particular benefits that we may observe across crawlers. There are two key dimensions in the assessment process. One key dimension is the nature of the crawl task. Crawl characteristics such as queries and/or keywords provided as input criteria to the crawler, user-profiles, and desired properties of the pages to be fetched (similar pages, popular pages, authoritative pages, etc.) can lead to significant differences in crawler design and implementation. The task could be constrained by parameters like the maximum number of pages to be fetched (long crawls versus short crawls) or the available memory. Hence, a crawling task can be viewed as a constrained multiobjective search problem. However, the wide variety of objective functions, coupled with the lack of appropriate knowledge about the search space, make the problem a hard one. Furthermore, a crawler may have to deal with optimization issues such as local versus global optima [1]. The other key dimension is to make comparisons and determine circumstances under which one or the other crawlers work best. Comparisons must be fair and be made with an eye toward drawing out statistically significant differences. Not only does this require a sufficient number of crawl runs but also sound methodologies that consider the temporal nature of crawler outputs. Significant challenges in evaluation include the general unavailability of relevant sets for particular topics or queries. Thus evaluation typically relies on defining measures for estimating page importance. However, literature review indicates that very limited work has been done to address this problem, i.e., to measure the page importance. It is imperative to measure the page importance to improve the crawler performance.

In order to deal with the above mentioned issue, this paper has proposed a new approach to measure the page importance and hence enhance the crawler accuracy. The inovation of this work is that the improved HMM has been introduce in measuring the page importance to enhance the crawler performance. The strong learning ability of the HMM has been fully used to evaluate the crawlers' page importance. By doing so, the data searching accuracy could be increased and thus the crawler performance could be enhanced. simulation tests have been implemented to verify the proposed approach. The analysis results have shown that the new model was very efficient and was superior to the existing Bayesian probabilistic model, Naïve Bayes model, and K-Nearest Neighbor Approach. Thus, the proposed improved HMM based approach has great impotance in business applications.

2. Research Method

When dealing with applications involving the display of advertisements in search engine results, the click-through rate (CTR) from independent Web users is important to measure and

thus determine the impact of the advertisement within the auction system that is being used. In this case, a stochastic approach can be used for modeling the user sessions [13].

For each website in the set of the similar topic that has the seeds' URLs in advance with the metasearch engine tool, we will start from the root URL of the website using the Viterbi algorithm, forecast the maximum probability link to get the similar topic pages on the basis of the root URL of each similar website and the trained HMM, and guide the crawler to crawl the target information to be collected [12]. We can obtain the most likely state sequence using the current observations and related parameters in HMM. If the number of pages in the similar websites and the target number of pages exceed the certain thresholds, they need a certain degree of control to improve the efficiency of collection by topic crawlers.

2.1. Generating a HMM Model

The topic learning process of the web crawler can be constructed well using HMM model, because the process of links from page to page is a hidden and unknown process, and distinguish the properties of a concrete web page is the dominant process which can be observed.

Constructing the HMM through simulating users' access sequences and analyzing the users' browsing mode, you can get the link structures between pages. According to the number of the pages access sequence, the topic-related pages are marked with hollow circles and the nontopic-related pages are marked with solid circles. The network diagram is constructed as following:

We can see from the definition of the HMM that the HMM is usually composed of two parts: state transfer model and the model from the state set to the observed sequences. Figure 1 shows the state transfer diagram.

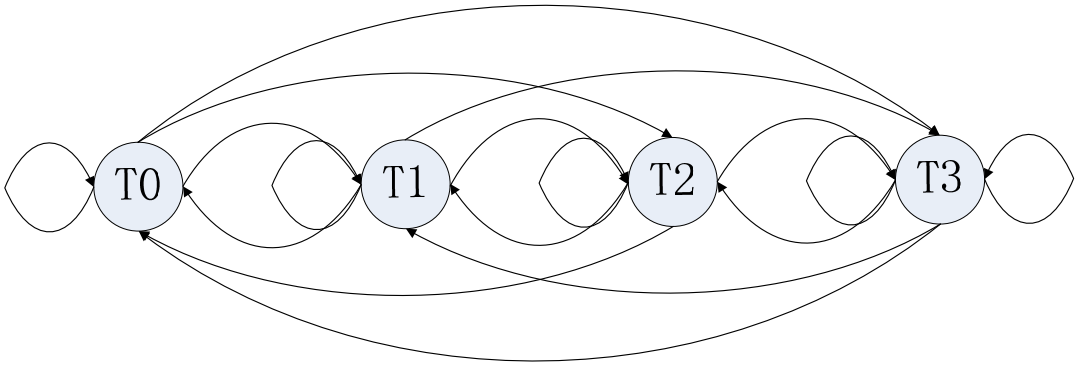

Figure 1. State transfer diagram

Imagining a web surfer who jumps from web page to web page, it chooses with uniform probability which link to follow at each step. The surfer will occasionally jump to a random page with some small probability. We consider the web as a directed graph. Fi be the set of pages which page i links to, and Bi be the set of pages which link to page i. After averaged over a sufficient number of steps, the probability of the surfer on page j at some point in time is given by the formula:

State transfer set $S = \{T_0, T_1 T_2, \ldots T_n\}$:

Observation set $O = \{O_1, O_2, \ldots, O_m\}$:

Parameter model of HMM with the known parameters of $\theta = (A, B, \pi)$.

First, we select the topic-related pages set T_3 as the target page set from the training set. T_i is the shortest distance that the page i from the target page T_3. As shown in Figure 2, ''4'' and ''7'' are the target pages. T_0 to T_3 are defined as following:

Second, for all pages we use the k-means algorithm for automatic clustering. Each category i am observed is corresponding to the observed value O_i.

Third, the initial probability distribution matrix is $\pi = \{P(T_0), P(T_1), \ldots, P(T_n)\}$, among which $P(T_i)$ represents the probability that the distance to the target topic page equals i in the initial state, where the initial values are generally evenly distributed, $\pi_i = P(q_1 = i)$ show time 1 choose the probability of a certain state. The transfer matrix is $A = \{a_{ij}\}$, among which a_{ij} indicates the probability that transferred from state Ti to state Tj . For example, if a_{30} equals 0, T_0 cannot be transferred from T_3 by only one step, and it is required at least three clicks to achieve the transformation. But when there are cross-links in pages, a_{jj}may be not equal to zero. The emission probability is $B = \{b_i(k)\}$, among which $b_i(k)$ indicate the probability in state T_k when the observed value O_i is known. Each emission probability b_{kj} indicate the probability from state S_k encountered the observed value *j*. $b_j(k) = P(v_k|j), 1 \le k \le M, 1 \le j \le N$, v_k indicate symbols. The symbols used in the following are as the standards in HMM.

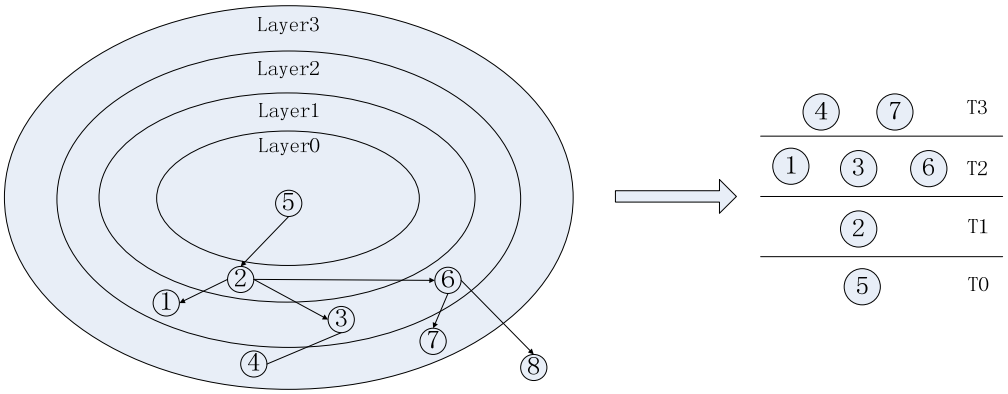

Figure 2. Network graph to state-relation diagram

2.2. Topic Model Training

The traditional HMM model reptiles to be further improved in dealing with the training set. First, the training set is selected by the expert web composed of related topics. And to mark the correlation with the theme of each page. If the small set, the workload is not great. However, if the collection is large, the work that tag is very heavy. Secondly, in the classification of the training set, equiring the user to determine the number of classes themselves. This is not only for the users is very difficult and inaccurate classification of reptiles crawl subsequent pages will have a significant impact on the process. Would affect the degree of reptiles crawl the web fundamentally.

On the basis of the above training set model, we use the Baurn–Welch algorithm. According to the estimated values of the initial given parameters, we keep on continuous iteration so that the various parameters tend to obtain more reasonable values.

A. Initialization

$\pi_i = r_i(i)$ shows the probability value of Si when t equals 1. In the model, it means the probability to reach the target pages when it step 1. Iterative calculation $\varphi(i,j)$ shows the probability at time t and t+1 with the state of S_i to S_j. $r_t(i) = \sum_{j=1}^{N} \varphi(i,j)$ show the probability at time t with the state of S_i.

$$\varphi(i,j) = \frac{P(q_t=i, q_{t+1}=j, (O|\theta))}{P(O|\theta)} \quad (1)$$

This showed the probability of the state of $S_j$at time *t* with the step of *i* and time *t+1*.

B. Iterative revaluation

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \varphi_t(i,j)}{\sum_{t=1}^{T-1} r_t(i)} \quad (2)$$

$$b_j(k) = \frac{\sum_{t=1,o_t=v_k}^{T} r_t(j)}{\sum_{t=1}^{T} r_t(j)} \quad (3)$$

We correct continuously for the state transition matrix and emission state matrix according to the values. $\sum_{t=1}^{T-1} r_t(i)$ indicate the number of times transferred from state S_i. $\sum_{t=1}^{T-1} \varphi_t(i,j)$ indicate the number of times that jumps from state S_i to state S_j

C. Termination condition

$$|log\ P(O|\theta) - log\ P(O|\theta_0) < \varepsilon| \quad (4)$$

Here, ε is the selected threshold in advance. $\log P(O|\theta_0)$ is the maximum probability of state sequence O after the reassessment of parameters iteration. In the HMM, it is the maximum probability of access paths of a URL. Continuous optimal correction of the parameters of the model makes the output parameters of the final training gradually move toward more optimum values.

2.3. Topic Learning Model

Let's look at two important assumptions HMM model:

Suppose one, when the state transition, the state transition probability at time *t* to *t +1* state transition time to the state when only the time t state, but not to the state before any moment. Can be expressed by the formula:

$$p(q_t = j|q_{t-1} = i) = p(q_s = j|q_{s-1} = i) \quad (5)$$

Meet assuming one , we say random sequence constitutes a first order Markov chain.

Assuming two, hidden Markov models assume that the output value, the output at time *t* is worth observing the current time *t* depends only on the probability in which the state has nothing to do with the previous state. Can be expressed by the formula:

$$b_j(k) = p(v_k|j) \quad 1 \le k \le M, 1 \le j \le N \quad (6)$$

In fact, these two assumptions are not used in the web between the very reasonable because the probability of observing the vector output appears at any one time depends not only on the current state of the system which, depending on the system and the time in which the previous state . Both assume that separates the relationship between pages and pages and pages relevant to the subject of the page or landing page orientation probability is large, then we have reason to believe that this website contains links to have a greater probability of target-oriented website . To be able to establish contact with the history of the state, the need to observation transfer the HMM's probability and output probability of state Hidden Markov assumption to make appropriate improvements.

Improved hidden Markov model assumes that the hidden state sequence is a second-order Markov chain. This state transitions is the state at time t to the state at the time *t +1l*, the state transition probabilities depend not only on the state at time t, and also depends on the state of the time *t-1*. Partial probabilities formula:

$$\begin{aligned} a_{ijk} &= P\left(q_{t+1} = s_k \middle| q_t = s_j, q_{t-1} = s_i, q_{t-2} = s_{i-1} \ldots .\right) \\ &= P\left(q_{t+1} = s_k \middle| q_t = s_j, q_{t-1} = s_i\right) \end{aligned} \quad (7)$$

$\sum_{k=1}^{n} a_{ijk} = 1, a_{ijk} \geq 0, 1 \leq i, j \geq n$, n in the model state number. Similarly, the probability of the current state of the output value not only depends on the system where the current state, and depends on the system before the moment of the state.

$$b_{ij}(l) = P(O_t|q_t = |S_i, q_t = S_i) \ \ 1 \leq i, j \leq n, 1 \leq l \leq m \tag{8}$$

In this paper, assumptions (7) (8) is presented on the basis of improved HMM learning algorithm is studied, it is concluded that the improved algorithm of forward algorithm and backward algorithm.

Forward and backward algorithm is calculated under the condition of θ given model to produce the probability of the observed sequence$O = O_1, O_2, \ldots, O_T$, is $\mathrm{P(O|\theta)}$.

By formula (7) indicated thatθ given model, the probability of θ certain state sequence $Q = q_1, q_2, \ldots, q_T$:

$$\begin{aligned} P(Q|\theta) &= P(q_1|\theta)P(q_2|q_1, \theta)P(q_3|q_1, q_2, \theta) \ldots P(q_T|q_{T-2}, q_{T-1}, \theta) \\ &= \pi_{q_1} a_{q_1 q_2} \prod_{t=3}^{T} a q_{t-2} q_{t-1} q_t \end{aligned} \tag{9}$$

Including π_i system at time $t = 1$ time state of S_i probability, the probability of a_{ij}state of said $S_i \rightarrow S_j$, that:

$$\begin{aligned} P(O|Q, \theta) &= P(O_1|q_1, \theta)P(O_2|q_1, q_2, \theta) \ldots P(O_T|q_{T-1}, q_T, \theta) \\ &= b_{q_1}(O_1) \prod_{t=2}^{T} b_{q_{t-1} q_t}(O_t) \end{aligned} \tag{10}$$

The formula (8) available in the state sequence Q (model has given) produced under the condition of the probability of observation sequence O, so to produce a given sequence in the context of the given model brother O probability:

$$\begin{aligned} \mathrm{P(O|\theta)} &= \textstyle\sum_{\mathrm{Q}} \mathrm{P(O, Q|\theta)} \\ &= \textstyle\sum_{\mathrm{Q}} \pi_{q_1} \mathrm{b}_{q_1}(o_1) \mathrm{a}_{q_1 q_2}(o_1) \prod_{t=3}^{T} \mathrm{a}_{q_{t-2} q_{t-1}} \mathrm{b}_{q_{t-1} q_t}(o_t) \end{aligned} \tag{11}$$

We can see that formula (11) to calculate $\mathrm{P(O|\theta)}$, its computation is very big, so the Forward and backward algorithm method is used to calculate as

$$\begin{aligned} \mathrm{P(O|\theta)} &= \mathrm{P}(o_1, o_2, \ldots, o_T = |\theta) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{P}\left(o_1, o_2, \ldots, o_t, o_{t+1}, \ldots, o_T, q_{t-1} = S_i, q_t = S_j \middle| \theta\right) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i, j) \beta_t(i, j), 2 \leq t \leq T - 1 \end{aligned} \tag{12}$$

3. Experiments and the Analysis

In order to evaluate the proposed HMM based crawler, numerical simulation tests have been carried out in this work. We simulated experiments for the seeds of the initial page performed by manual. The relevant page of the theme page, the seeds of the seeds of each theme page set size of 100. For each topic, the spiders crawling to the page and the result of the seed, because each page of the crawler return, their document similarity is obtained by using the method of VSM, if their similarity values of maximum value is bigger than a user-defined threshold, then the page will mark the results so far. Check the result of the crawler, the more the crawler is successful, the spiders crawling to the theme and the probability of the similar result is higher. The performance of the crawler is the average of all the theme of the check results.

Figure 3 shows the simulation test results, where the performance of the proposed improved HMM has been compared with the original HMM algorithm.

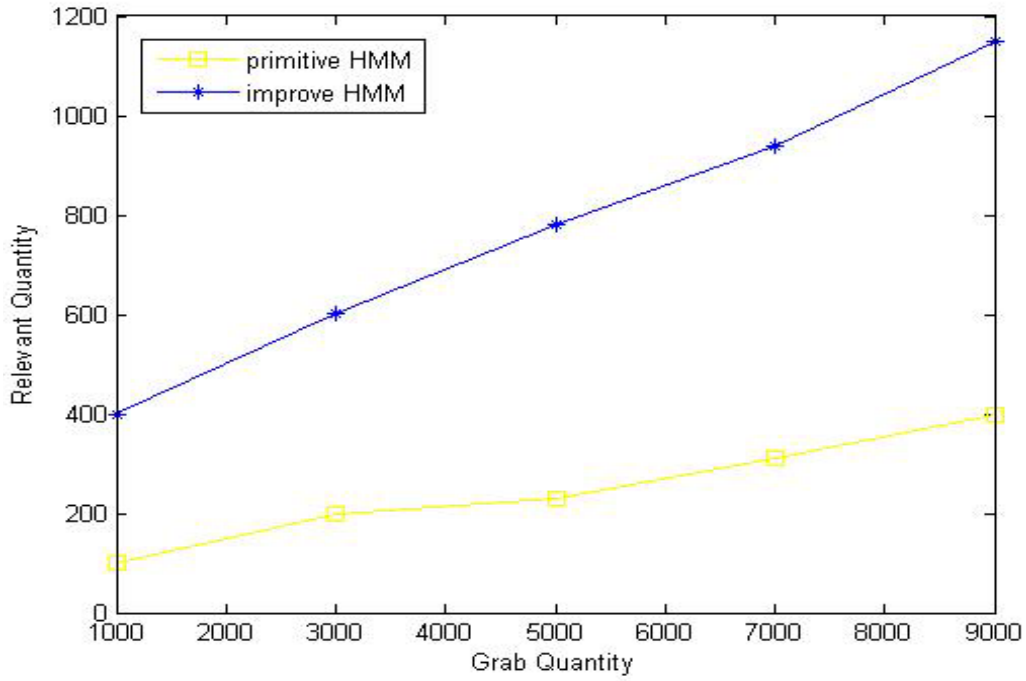

Figure 3. The simulation results

It can be see in Fig. 3 that the improved HMM obtains the grab quantity of 8997 with 1152 relevant themes while the primitive HMM only get 400 relevant themes. Hence, the improved HMM outperforms the primitive HMM.

Table 1 lists the comparison of the proposed method against the existing Bayesian probabilistic model, Naïve Bayes model, and K-Nearest Neighbor Approach.

Table 1. The comparison results of the crawler searching

Method	Grab quantity	Relevant quantity
Bayesian probabilistic model	8654	981
Naïve Bayes model	8549	1005
K-Nearest Neighbor	8733	973
Improved HMM	8997	1152

One can note in table 1 that the Bayesian probabilistic model obtains the grab quantity of 8654 with 981 relevant themes; the Naïve Bayes model obtains the grab quantity of 8549 with 1005 relevant themes; the K-Nearest Neighbor obtains the grab quantity of 8733 with 973 relevant themes. Hence, the performance of the proposed HMM based approach is among the best one. This results indicate that the improved HMM could provide effect measurement of page importance. As a result, the crawler searching performance is better than the other methods. Hence, the comparison results indicate that the proposed HMM based crawler can improve the information retrieval.

4. Conclusion

In order to improve the web crawler performance, this work proposed a new model based on the hidden markov (HMM) chain. Through numerical simulation test, the analysis results have shown the proposed algorithm was effective to guide the focused crawling based on the user path in HMM. The accuracy of data acquisition will be higher if the recognition model that the topic relies on can continue to iterate optimization in theory. The analysis results indicate that the HMM topic crawler has a good prospect in enhancing the web crawler

performance. Future work will verify the proposed crawler search system based on HMM in business applications.

Acknowledgements

This work was supported by Chinese College Students' Innovative Entrepreneurial Training Program (No. 201210500024).

References

[1] Mobasher B, Dai H, Luo T, Nakagawa M. *Effective personalization based on association rule discovery from web usage data*. In Proceedings of the 3rd International Workshop on Web Information and Data Management, WIDM 2001: 9–15.
[2] Chakrabarti S, Berg M, Dom B. *Focused crawling: a new approach to topic-specific Web resource discovery.* In Proceedings of the 8th International WWW Conference. 1999: 237-252.
[3] Pant G, Srinivasan P. Learning to crawl: Comparing classification schemes. *ACM Transactions on Information Systems*. 2005; 23: 430-462.
[4] Kollerand D, Sahami M. *Hierarchically classifying documents using very few words*. In Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97). 1997: 170-178.
[5] Wang W, Chen X, Zou Y. *A focused crawler based on naïve bayes classifier.* In Proceedings of the Third International Symp. on Intelligent Information Technology and Security Informatics, China. 2010: 517-521.
[6] Yang Y, Lui X. *A reexamination of text categorization methods.* In Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99). 1999: 42-49.
[7] Dixit A. *Design of Scalable Parallel Migrating Crawler Based on Augmented Hypertext Documents. Ph.D. Thesis*. MDU. 2010.
[8] Page L, Brin S, Motwani R, Winograd T. *The pagerank citation ranking: bringing order to the web.* Technical Report, Stanford InfoLab. 1998.
[9] Xing W, Ghorbani A. *Weighted pagerank algorithm.* In Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04). 2004: 567-578.
[10] Ding C, He X, Husbands P, Zha H, Simon H. *Link analysis: Hubs and autorities on the world.* Technical Report. 2001: 447-463.
[11] Kelly D, Teevan J. *Implicit feedback for inferring user preference: A bibliography*. In SIGIR Forum. 2003: 521-536.
[12] Tan Q, Mitra P. Clustering-based incremental web crawling. *ACM Trans. Inf. Syst.* 2010; 28: 4-18.
[13] Sadagopan N, Li J. *Characterizing typical and atypical user sessions in clickstreams.* In Proceedings of the 17th International Conference on World Wide Web. 2008: 885–894.
[14] Wang X, Liu C. Semantic representation of complex resource requests for service-oriented architecture. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2014; 12(1): 741–746.
[15] Hermawan H, Sarno R. Developing distributed system with service resource oriented architecture. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2012; 10(2): 389–399.
[16] Paulus I. Cost and benefit of information search using two different strategies. *TELKOMNIKA*. 2010; 8(3): 195-206.
[17] Qu X, Wang Y. The research on software resource re-sharing for small and medium-sized enterprise cloud manufacturing system. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(1): 711-717.