International Journal of u- and e- Service, Science and Technology
Vol.9, No. 8 (2016), pp.227-236
http://dx.doi.org/10.14257/ijunesst.2016.9.8.19

Design of Web Crawler Based on Improved Hidden Markov Model

Hailong Jia[1] and Lina Fang[2]

[1]*Wuhan University of Technology, Wuhan 430063, China*
[2]*Xinxiang University, Xinxiang 453000, China*
[1]*47160216@qq.com*, [2]*501597924@qq.com*

Abstract

This paper analyzes the shortcomings of traditional hidden Markov crawler, makes some improvements on the clustering strategy of web pages and the judgment algorithm for determining the correlation of pages or hyperlinks with the topic; and brings forward an AHMM (Adaptive Hidden Markov Model) modeling method. The experimental results shows that the improved AHMM is much more efficient than the traditional HMM.

Keywords: *net-mediated public sentiment; hidden Markov; web crawler*

1. Introduction

The collection of net-mediated public sentiment is to collect accurate, completed and well-timed comments on a social hot event released by internet users. The social hot event can be considered as the theme information users concerned. Therefore, the collection of net-mediated public sentiment can be achieved by means of the theme crawler. The key point of theme crawler is to estimate the correlation of new URLs from crawled pages. Its performance depends on the ability of the model to analyze and forecast web content. The statistical model, like Hidden Markov Model (HMM), has a great advantage in simplification and language accuracy [1]. This paper analyzes and summarizes the HMM's shortcomings in collecting public sentiments, and solves those problems from three aspects—the training clustering strategy, topic identification method and modeling—to improve the traditional HMM.

2. The Shortcomings of Traditional HMM Crawler

To compare traditional HMM theme crawler with the classic Best First (BF) theme crawler on content strategy BF (Content BF), link strategy BF (Link BF) and composite strategy BF (Composite BF), we get the results from comparison test which are presented in Figure 1.

Figure 1, is a precision ratio comparison of crawlers. As is shown in Figure 1, the precision ratio of composite strategy BF crawler, the content strategy BF crawler and the link strategy BF crawler are respectively 16.6%, 13.2% and 5.0%. However, the precision ratio of traditional HMM theme crawler is only 3.1%, lower by about 13.5% than composite strategy BF crawler.

For the above shortcomings, the traditional HMM crawler needs to get corresponding improvements to increase precision and reduce the running time.

ISSN: 2005-4246 IJUNESST

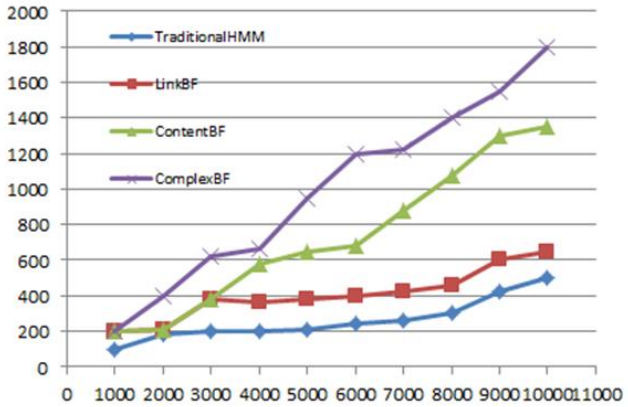


Figure 1. Comparison Chart of Traditional HMM Crawler and BF Crawler

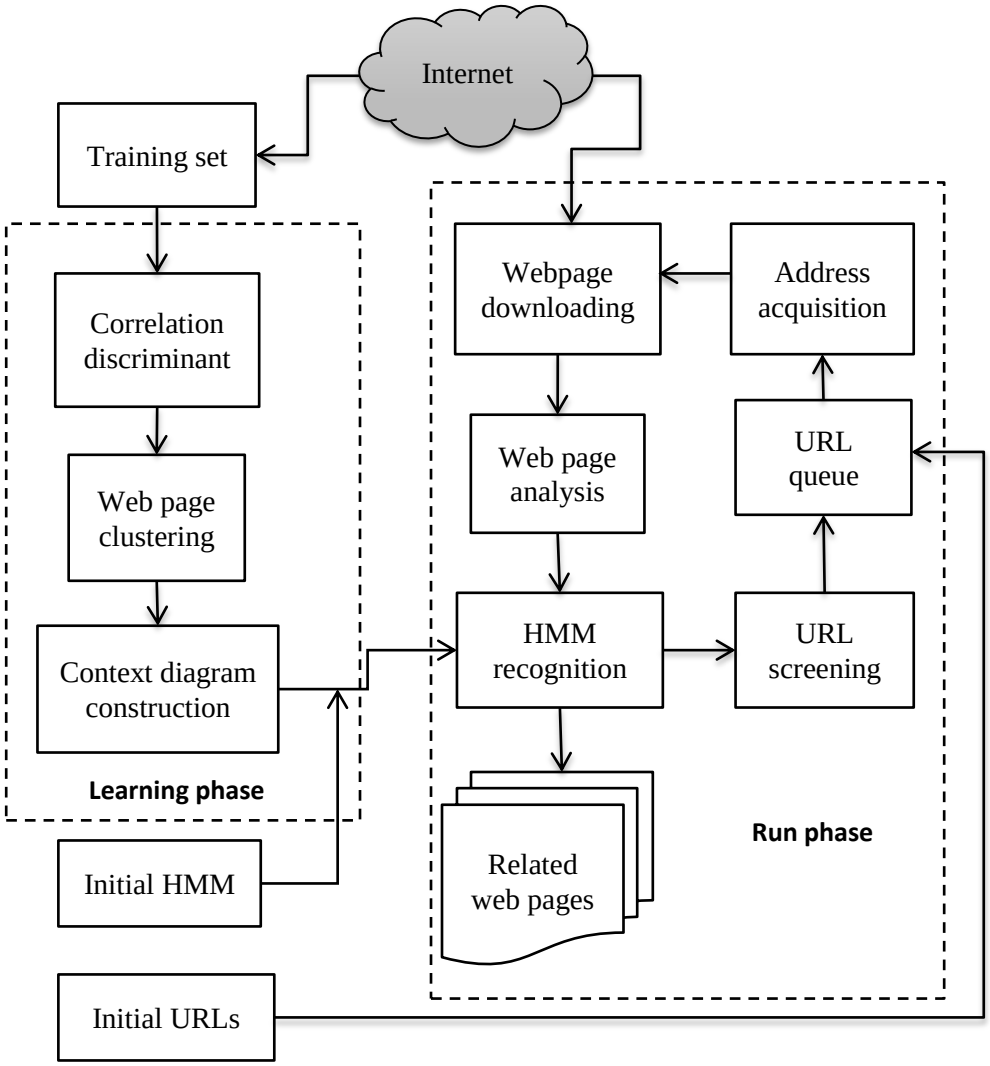


Figure 2. The Framework of HMM Crawler

3. The Overall Framework of HMM Crawler

Hidden Markov model is used to describe a Markov process which has connotative unknown parameters. This means that firstly the implicit parameters of

the process will be determined from the known parameters, and then further analysis will be made by using these parameters. After the HMM is introduced to the theme crawler, the recognition process of HMM will be improved, so we can use the trained HMM to predict the crawling path. The framework of HMM theme crawler is shown in Figure 2.

There are three stages in the operation of HMM crawler:

(1) The initial stage, in which the relevant parameters are set, the initial training set of the initial page is selected, and the page is pretreated.

(2) The learning stage, in which the web page is optimized by HMM training including correlation estimation, clustering and construction of context diagram.

(3) The running stage, in which the related sites of the theme are looked up, then using the optimized HMM to analysis the page and hyperlink of the sites, and storing the analyzed information in the corresponding list, and then downloading the corresponding page data to the local according to the path which is recognized by theme target.

4. Training Set Page Clustering Strategy of AHMM Crawler

K in K-means algorithm is given by the users, so when the traditional HMM crawler is in dealing with the training set, the users need to determine the classification number K of training set. This must be a high requirement for the users, and at most of the time, it is difficult for the users to determine how many categories should the training set be divided into, and moreover, the size of the K value will fundamentally affect the efficiency and accuracy of crawlers. So it is necessary to improve the obtainment of K value. At present, there are many studies on obtainment and optimization of K value, for example, some experts get and optimize the K value in K-means algorithm by using the knowledge of distance cost function, genetic algorithm, histogram, graph theory and so on [2, 3, 4]. This paper is intended to get K value automatically by using the method in reference [5] and avoid many unreasonable factors brought by user-defined K value, thereby improving the precision of HMM crawler. Automatic acquisition of k values in k-means algorithm is as follows:

Input: *T* sample data;

Output: *R, Z, r, m, h;*

Step1: Extract *T* data from the data to be clustered as sampling data of clustering parameters;

Step2: Calculate the distance $d(x_i, x_j)$ between the sampling data to obtain a diagonally symmetrical distance matrix *D*. Each column represents data from the one with all the sampling data;

Step3: Find the minimum distance $min\ d(x_i, x_j)$;

Step4: Calculate the average value R_i of each column in distance matrix *D*, and then take the average of R_i to obtain $\bar{R}$;

Step5: According to the formula $\frac{|R_i - \bar{R}|}{\bar{R}}$ remove noise points with error greater than 30%, enabling the remaining points as belonging to the same cluster;

Step6: Recalculate $\bar{R}$, according to $\bar{R}$ and $min\ d(x_i, x_j)$, to obtain high density radius $R = \bar{R} + min\ d(x_i, x_j)$;

Step7: In the column $min\ R_i$ resided, according to $d(x_i, x_j) <$ min R_i, get the number *Z* of high density;

Step8: According to *R* and $min\ d(x_i, x_j)$, cluster partition radius $r = R + min\ d(x_i, x_j)$, the distance between the center of the combined cluster $m = r + r$, the

distance between the boundary points of the combined cluster $h= min\ d(x_i, x_j)+ min\ d(x_i, x_j)$.

Wherein, *R* represents the high density radius, *Z* represents the number of high-density, *r* represents the cluster partition radius, *m* represents the maximum distance between the center of the two terms of the merger between the clusters, *h* is the maximum distance between the boundary points.

5. The Selection Method of AHMM Crawler for the Page to be Collected

The selection criterions of the page to be collected is correlation degree. The discrimination task of topic correlation is to deliver the page after pre-treatment to the relevant calculation module to work out the value of the correlation index and to determine whether the page information associated with the topic. Considering the web designers tend to aggregate levels according to topic and the hyperlinks in page usually has a relationship with or similar to that topic, therefore, in this paper, we integrated the page content and link structure, proposed the correlation identification method of combining content and links, and then, by analyzing link structure in pages and using the relevant themes weighting table to measure the correlation between page content and links, estimated the degree of correlation between the page and the topic.

5.1. Judgment on the Correlation Degree Between the Page and the Topic

After the preparatory work, there is a judgement on correlation degree of the topic. At first, the correlation degree between the topic and the page should be analyzed. Seed URLs are the pages given in advance which are highly relevant to topics. The crawlers collect seed page first and analyze its correlation degree. As is known to all, Web page is somewhat different from traditional plain text. Web page is a semi-structured data, which includes text pages, hyperlinks, labels and so on. Among them, the web page content is pure text. A typical HTML source code of web page is as follows:

<html>
<head>
<title> page title</title>
</head >
<body>
body content
</body>
</html>

So in the analysis of topic correlation degree, the relevance of text information of the page will be calculated firstly. If the text information of page has a strong correlation with the topic, this page's topic will correlate with the page. When analyzing the textual information, we firstly use ICTCLAS word text message preprocessing to do word processing in order to effectively deal with Chinese web page, then, according to equation (1), calculate the correlation degree of text information. In equation (1), D_1 represents the page which has been accessed, D_2 is the keyword of theme, and W_{ik} indicates the weights of corresponding characteristics.

$$Sim(D_1, D_2) = \sum_{k=1}^{N} W_{1k} * W_{2k} \quad (1)$$

According to the vector space model (VSM), the given natural language document D can be expressed as $D = (t_1, w_1 ; t_2, w_2 ; \ldots\ldots ; t_n, w_n)$, where t_i

represents the features and w_i indicates the weights of the features in the document, and weights are generally calculated on the basis of frequency of the features. But it's a little difficult to analyze the features because they not only can recur in the document but also have precedence. To simplify the analysis, we will not consider the order of features in the document and request mutually different features. Then you can take features $(t_1, t_2, \ldots\ldots t_n)$ as an n-dimensional coordinate system, and the weights $(w_1, w_2 \text{ ; } \ldots\ldots \text{ ; } w_n)$ as the corresponding coordinate values, so a document is represented as a vector which has n-dimensional space, called $D=(w_1, w_2 \text{ ; } \ldots\ldots \text{ ; } w_n)$ as a vector representation or vector space model of the document *D*. The similarity (Similarity) of two documents D_1 and D_2 will be measured by $Sim(D_1, D_2)$.

In addition to text information in a Web page, the label information which need to be concerned about are as the followings: Page Title <TITLE>, which outlines and summarizes the contents of the entire page, playing a key role; the font with particular emphasis <STRONG> reflects the information on which the authors emphasize; and others which play a local modification, more or less emphasis on the part of the content of the page, such as: Bold <B>, italic <EM>, Underline <U>, italic <I> and so on. According to the different important degree reflected by label information, this paper provides a chart of weight value of some of the HTML labels, as is shown in Table 1.

Table 1. The Weight Value of HTML Labels

Numble	Label	Weight
1	<Title>	1
2	<H1>	0.9
3	<H2>	0.7
4	<H3>	0.6
5	<Strong>	0.5
6	<B>	0.4
7	<EM>	0.4
8	<U>	0.4
9	<I>	0.4

Taking the text and labels of the website into consideration, the correlation degree W(p) of page P to the topic is shown in equation (2) below:

$$W(p) = W_0(\text{p}) + \sum_{i=1}^{N} \lambda_i * W_i(\text{p}) \quad (2)$$

Wherein, $W_0(\text{p})$ indicates topic correlation degree of the text information of page *P*, which is calculated by equation (1); *N* is the number of labels in the corresponding label and weight table; λ_i represents the corresponding weights of NO. When it is *i* ; $W_i(\text{p})$ represents the correlation degree between the corresponding text information of tag *i* and the topic , which is also calculated by the formula (1).

5.2. Judgment on the Correlation Degree Between URLs and the Topic

There are a lot of hyperlinks in web pages. It is generally believed that the use of these hyperlinks can help link to other relevant information layer by layer and dig further. Before connecting the analysis, we need to remove noise connection, for example, next link, advertising links, page navigation links, *etc*..

In analyzing the links, at first you need to determine the page type, extract links from qualifying pages, and carry out necessary conversions to URLs. Through determining the relevance of URLs and the theme, you can improve the collection

rate of effective pages. Sometimes encountered label with a link, such as <A> label, you need to extract the anchor text (Anchor). The link extraction process in page is as follows:

Link extraction algorithm:

Input: page file

Output: URLs in the page, anchor text of URLs (Anchor)

Step l: take page file, if the page file queue is empty, turn into Step 6;

Step 2: judge page types, if it does not conform to this abandoned file, turn into Step l;

Step 3: sequentially read a file, abandoning the noise links, extracting the URL and anchor text in the label with link, when meet the terminator of file, turn into Step 6;

Step 4: adjust the format of the extracted URLs;

Step 5: store extracted URLs and its error text, turn into Step l;

Step 6: end, link extraction completed.

Before calculating the similarity of URLs, you need to prepare for: firstly, calculating the correlation between the downloaded pages and themes and saving the calculated correlation value and corresponding URL address; next, setting up the "high correlation chain", this paper suggests that the first 50 URLS with high correlation value is high related; at last, evaluating comprehensively the correlation between L and themes from the four aspects, including the superior page P of URL (denoted L) which needs to be collected, wrong text of L, the URL relevance of L and the distance between L and "high correlation chain" .

6. HMM Modeling in AHMM Crawlers

The crawling way of traditional theme crawler usually goes in this way: When the crawler fetches a page, it first determines whether the page is relevant to the theme, and if it is relevant, the crawler will crawl the web page and extract the links in it; if not relevant, it will stop. However, there is one major drawback that there are also links pointing to the page theme in some irrelevant pages which had been judged by theme crawler, and this is the so-called tunnel problem encountered in the process of crawling [6]. It makes crawler lost some indirect information related with the subject. According to some studies, it was found that in the pages crawled by traditional HMM crawler, the relevance of topic is not too high, but the crawling time is much longer. This must be correlated with the unreasonable preconditions and algorithm of HMM.

There are a lot of space correlation between the pages. In order to establish contact with the historical state, the paper makes appropriate modification to two HMM assumptions. The improved HMM hidden state sequence assumption is as follows: When it transfers from the state of time *t* to time *t+1*, its transition probability depends on the state of time *t* and *t-1* , as is shown in equation (3):

$$a_{ijk} = P(q_t = k | q_{t-2} = i, q_{t-1} = j) \quad (3)$$

Similarly, the probability of the output value of the current status depends on the present state and the state of the system at last moment, as is shown in equation (4):

$$b_{ij}(k) = P(o_t(k) | q_{t-2} = i, q_{t-1} = j) \quad (4)$$

Based on improved assumptions, the learning algorithm of HMM also needs to be modified:

(1) Forward Algorithm

At first, Forward variable ∂ can be defined, which is shown in formula (5):

$$\partial_t(\mathrm{i,j}) = \mathrm{P}(O_0, \ldots, O_i, q_{t-1} = i, q_t = j|\lambda) \quad (5)$$

Wherein $\partial_t(i,j)$ shows that in the model λ, when the time is *t*, the probability's partial observation sequence is $P(O_0, \ldots, O_i)$. And Forward variable $\partial_t(i,j)$ can be recursively solved as follows:

Initialization: $\partial_t(i,j)=\pi_i \mathrm{b}_i(O_0)a_{ij}b_{ij}(O_t), 0 < i,j < N-1;$

Iteration: $\partial_t(i,j) = \left[\sum_{k=0}^{\mathrm{N}-1} \partial_{t-1}(i,j)a_{ijk}\right]b_{jk}(O_t), 1 < t < T-1, 0 < i,j < N-1;$

End Condition: $\mathrm{P(O}|\lambda) = \sum_{t=0}^{N-1} \partial_{t-1}(i,j)$

(2) Backward Algorithm

Similarly, backward variable β can also be defined, as is shown in equation (6):

$$\beta(\mathrm{i,j}) = \mathrm{P}(O_{t+1}, O_{t+2} \ldots, O_{t-1}|\mathrm{q}_{t-1} = i, q_t = j, \lambda) \quad (6)$$

Wherein, *1<t<T-1*, *0<i*, *j<N-1*, and Backward variable *β(i,j)* can be solved recursively as follows:

Initialization: $\beta_{\mathrm{T}-1}(i) = 1, 0 < i,j < N-1;$

Iteration: $\beta_t(i,j) = \sum_{k=0}^{\mathrm{N}-1} \mathrm{a}_{ijk}\, b_{jk}(O_{t+1})\beta_{t+1}(j,k), t = T-2, T-1, \ldots, 0, 0 < i,j < N-1;$

End Condition: $\mathrm{P(O}|\lambda) = \sum_{i=0}^{\mathrm{N}-1}\sum_{j=0}^{N-1} \partial_{t-1}(i,j)\beta_{t-1}(i,j), 1 \le t \le T.$

According to Forward-Backward algorithm, in the case of the model λ, the probability of the generating observed sequence is: $\mathrm{P(O}|\lambda) = \sum_{i=0}^{\mathrm{N}-1}\sum_{j=0}^{N-1} \partial_{t-1}(i,j)\beta_{t-1}(i,j), 1 \le t \le T.$

7. Implementation and Experimental Analysis

7.1. The HMM Training

The initial set of HMM model probability cannot be directly applied. We need to train the model and make the model tend to be mature. So we can train the concrete structure of concentrated learning model through context diagram and to build a maximum specific model through the training data. The detailed process is as follows:

(1) Firstly, initialize the data, given seeds URLs which are closely integrated with the theme and the page which point to can be taken as the zeroth layer of the context diagram;

(2) Secondly, we can reversely collect all the pages which point to the zeroth layer by using general search engine. In this step, the pages which reversely linked constitute the 1st layer of page set;

(3) And so on, an N layer context diagram is forming, in which the layer number represents the distance and indirectly reflects the similarity.

After that, we can use the structured context diagram to model. And the distance to target page, *i.e.*, the level which page belongs to is the state of the HMM. Then, the dynamic clustering was carried out on the page. Then, the different clustering categories between different topic relevance characteristic vector can be taken as the observation value in the HMM model.

7.2. The HMM Path Prediction

After the HMM training has been completed, this paper predicted the collected page based on the HMM model and identified the optimal sequence of page collection path by using the Viterbi algorithm.

In the process of collection, this paper used the Viterbi algorithm in the theory of HMM to guide the crawler to crawl. After the page which URL points to has been

downloaded, it needs to analyze and extract the URL of the page, store the URL and anchor text in the database, and then build the context diagram, and finally calculate the distance from the page to the target topic according to the layer of the page and distribute the priority in terms of the distance. This paper is trying to use the current observations and related parameters of HMM to deduce the most likely state sequence and predict the path of the target page.

7.3. Experimental Analysis

Compared the AHMM crawler designed by this paper with the traditional HMM crawler and the classic Best First crawlers: content strategy BF crawlers, composite strategy BF crawlers, we can get the results from the comparison tests, which is shown in Figure 3.

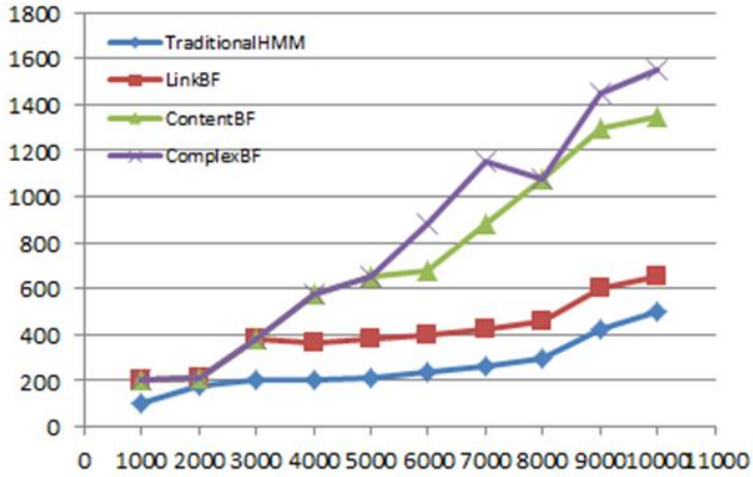


Figure 3. The Comparison of Crawler's Precision Rate

Figure 3, is a comparison of crawler's precision rate. In this figure, we can find that the precision rate of composite strategy Best First crawler is 15.5%, and that of content strategy Best First crawler is 12.8%, while the precision rate of AHMM crawler is 15.2%, and the rate of traditional HMM crawler is only 3.3%.

Table 2, shows the average crawling time and precision rate of these crawlers crawling 10,000 pages. The results show that the performance of AHMM crawler in this design is better than traditional HMM crawler, but it did not get a good effect compared with the composite strategy Best First crawler, which mainly due to a large number of AJAX pages in the obtained seed URL. We can also find that whether the AHMM crawler in this paper or the classic Best First crawler, it is more difficult for most of which to crawl the dynamic information in AJAX pages, and this resulting non-ideal precision rate.

Table 2. Run Time and Precision Rate of Crawlers

Crawlers	Time (min)	Precision Rate (%)
Composite strategy BF	1271.	15.5
Content strategy BF	738.7	12.8
AHMM	1387.4	15.2
Traditional HMM	1404.2	3.3

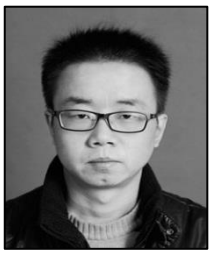
8. Conclusion

Based on the traditional HMM crawlers, we made several improvements. First, we made improvements on selecting *K* values in *K-means* algorithm of the training clustering strategy to avoid problems caused by the user-defined; Secondly, we improved the method of determining the degree of correlation; thirdly, for the problems that the correlation of HMM crawler is not high and some indirectly related pages can easily be lost, we modified assumptions of HMM to improve its modeling methods and put forward improved respective Forward-Backward; Finally, by using the improved method, an AHMM crawler has been produced, and the experimental results show that the improved AHMM is much more efficient than the traditional HMM.

References

[1]. L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models", IEEE ASSP Mag, **(1986)**, pp. 4-16.
[2]. M. Stamp, "A Revealing Introduction to Hidden Markov Models", **(2004).**
[3]. J. Picone "Continuous Speech Recognition using Hidden Markov Models", IEEE Assp May. vol. 7, no. 3, **(1990)**, pp. 26-41.
[4]. L. Rabiner and B. Juang, "An introduction to hidden Markov models", ASSP Magazine JEEE, vol. 3, no. 1, **(1986)**, pp. 4-16.
[5]. H. Liu, J. Janssen and E. Milios, "Using HMM to learn user browsing patterns for focused web crawling", Data&Knowledge Engineering, vol. 59, no. 2, **(2006)**, pp. 270-329.
[6]. D. Bergmark, C. Lagoze and A. Sbityakov, "Focused Crawls, Tunneling, and Digital Libraries", In Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries", **(2002)**, pp. 91-106.

Authors



Hailong Jia (1982-), male, nationality: Han; place of birth: Baoji, Shaanxi; lecturer of Computer Science and Technology in Modern Educational Technology Center of Xinxiang University; educational background: master's degree of Computer Application Technology in Beijing Industry University in 2010, doctor's degree of Information and Communication Engineering in Wuhan University of Technology (PhD Candidate); research directions: image recognition, network technique, information and communication engineering;

Lina Fang (1980-), female, nationality:Han; place of birth:Zhoukou,Henan; College of Computer and Information Engineering in Xinxiang University; educational background: Master's degree of Computer application technology in Huazhong University of Science and Technology in 2009; research directions: Computer application,multimedia information processing, image transmission and processing.