An Improved Hidden Markov Model for Literature Metadata Extraction

Bin-Ge Cui and Xin Chen

College of Information Science and Engineering,
Shandong University of Science and Technology, 266510 Qingdao, China
cuibinge@yahoo.com.cn

Abstract. In this paper, we proposed an improved Hidden Markov Model (HMM) to extract metadata in the academic literatures. We have built a dataset including 458 literatures from the VLDB conferences, which contains the visual feature of text blocks. Our approach is based on the assumption that the text blocks in the same line have the same state (information type). The assumption is effective in more than 98% occasions. Thus, the state transition probability among the same states in the same line is much larger than that in different lines. According to this conclusion, we add one state transition matrix for HMM and modified the Viterbi algorithm. The experiments show that our extraction accuracy is superior to that of any existing works.

Keywords: HMM, Viterbi; Metadata Extraction, Text Block.

1 Introduction

With regard to the issue of metadata extraction, there have been three main approaches, i.e., heuristic approach [1], template-based approach [2], HMM-based approach [3]. Among these approaches, heuristic method has a low accuracy (only about 80%); template-based approach has a high accuracy (about 95%), but it requires manually writing template, thus it has poor adaptability; HMM-based approach has a medium accuracy (about 87%). It requires a large training sample to construct the transition matrix and emission matrix. However, to further improve the accuracy of HMM-based approach is very difficult.

In this paper, we proposed to add the text block visual features into the HMM. As we know, humans recognize the various parts of a PDF literature not just relying on the content of the text blocks, but also on their location, size, font and other information. In general, the text blocks in the same line is very likely to belong to the same information type, i.e., title, author, affiliation, etc. However, existing data sets do not provide the text block location information [4]. To solve this problem, we downloaded about 500 literatures from VLDB conferences and converted them into HTML format using an open-source utility [5]. The converted HTML document contains the text block location and other font information. Based on location information, we divide the state transition matrix of HMM into two transition matrixes. One matrix represents the state transition probability in the same line; the other represents the state transition probability in different lines. We modify Viterbi algorithm with the

D.-S. Huang et al. (Eds.): ICIC 2010, LNCS 6215, pp. 205–212, 2010.

two matrixes. Experimental results show that the exaction accuracy increases 4.5%, and the overall metadata exaction accuracy reaches 97.4%.

2 PDF File Conversion

2.1 HMM Profile

HMM is a double stochastic process, which can be defined as a five-tuple $\lambda=(X, O, \pi, A, B)$. X denotes a set of states, $X =\{S_1, S_2,..., S_N\}$, N denotes the state number, q_t denotes the state at time t; O denotes a set of observed symbols, $O= \{V_1, V_2, \ldots, V_M\}$, M denotes observed symbols number; π denotes the initial state distribution, $\pi = \{\pi_i\}$, $\pi_i = P\{q_1 = S_i\}$, $1 \le i \le N$; A denotes the state transition probability distribution, $A = \{a_{ij}\}$, $a_{ij} =\{q_{t+1} = s_j \mid q_t = s_i\}$, $1 \le i, j \le N$; B denotes the observation probability distribution of state j, $B = \{b_j(k)\}$, $b_j(k) = P\{O_t=V_k \mid q_t = S_j\}$, $1 \le j \le N$, $1 \le k \le M$. The detailed definitions can be found in reference [6].

2.2 Conversion of PDF Literature into HTML Files

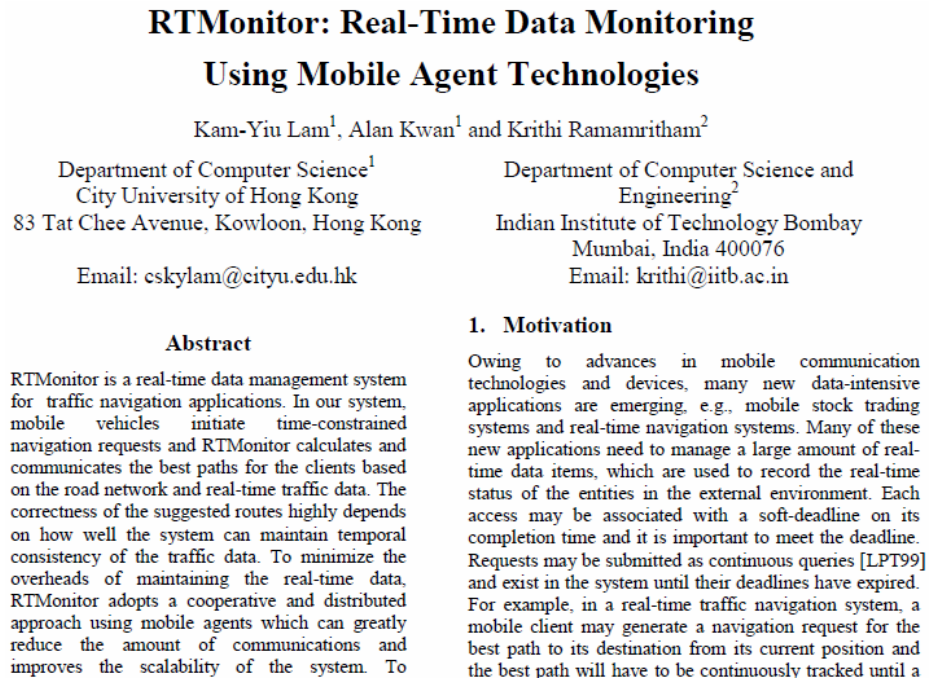
In this section, we provide a sample PDF literature, which is shown in Figure 1. As we have seen, it contains title, authors, affiliations, addresses, emails, abstracts, etc. These items are called the literature metadata. They play an important role for the readers to understand a summary of the literature information. However, writing a program directly that can analyze the contents of PDF literature is very difficult. HTML documents are relatively easy to analyze and process. We utilize the open source Pdftohtml to convert PDF literatures into HTML format. The converted result is shown in Fig.2. From the 3rd line to the 25th line, each text block has its own location information.

3 HTML Document Pre-processing and Conversion

Many parts of the HTML document are not necessary for our follow-up metadata extraction, e.g., the font information, the background image, etc. Therefore, we need to pre-process the document collection, and then integrate all the HTML documents into an XML document. Pre-processing steps are as follows:

Step 1: Positioning the file pointer to the first position of a **DIV** appears;
Step 2: Loop read each line, and assign the content to the variable ***line***;
Step 3: Parse out the **top** and **left** property values in the **DIV** element's **style** attribute, and assign them to the variables ***vtop*** and ***vleft*** respectively;
Step 4: Parse out the text block content in the **span** element of ***line***, and reassign it to ***line***;
Step 5: Remove the ****, ****, ****, **, <a>, **, **
** tags and some other meaningless symbols from ***line***;

After the pre-processing steps, one DOM tree will be created, which contains the type (title, author, etc.) and location for all text blocks in all literatures.



RTMonitor: Real-Time Data Monitoring Using Mobile Agent Technologies

Kam-Yiu Lam[1], Alan Kwan[1] and Krithi Ramamritham[2]

Department of Computer Science[1]
City University of Hong Kong
83 Tat Chee Avenue, Kowloon, Hong Kong

Email: cskylam@cityu.edu.hk

Department of Computer Science and Engineering[2]
Indian Institute of Technology Bombay
Mumbai, India 400076
Email: krithi@iitb.ac.in

Abstract

RTMonitor is a real-time data management system for traffic navigation applications. In our system, mobile vehicles initiate time-constrained navigation requests and RTMonitor calculates and communicates the best paths for the clients based on the road network and real-time traffic data. The correctness of the suggested routes highly depends on how well the system can maintain temporal consistency of the traffic data. To minimize the overheads of maintaining the real-time data, RTMonitor adopts a cooperative and distributed approach using mobile agents which can greatly reduce the amount of communications and improves the scalability of the system. To

1. Motivation

Owing to advances in mobile communication technologies and devices, many new data-intensive applications are emerging, e.g., mobile stock trading systems and real-time navigation systems. Many of these new applications need to manage a large amount of real-time data items, which are used to record the real-time status of the entities in the external environment. Each access may be associated with a soft-deadline on its completion time and it is important to meet the deadline. Requests may be submitted as continuous queries [LPT99] and exist in the system until their deadlines have expired. For example, in a real-time traffic navigation system, a mobile client may generate a navigation request for the best path to its destination from its current position and the best path will have to be continuously tracked until a

Fig. 1. One Typical PDF Literature

```
<BODY bgcolor="#A0A0A0" vlink="blue" link="blue">
<IMG width="892" height="1263" src="371001.png" alt="background image">
<DIV style="position:absolute;top:133;left:217"><nobr><span class="fc0"><b>RTMonitor: Real-Time Data Monitoring  </b></span></nobr></DIV>
<DIV style="position:absolute;top:172;left:259"><nobr><span class="fc0"><b>Using Mobile Agent Technologies </b></span></nobr></DIV>
<DIV style="position:absolute;top:219;left:252"><nobr><span class="fc1">Kam-Yiu Lam</span></nobr></DIV>
<DIV style="position:absolute;top:216;left:354"><nobr><span class="fc2">1</span></nobr></DIV>
<DIV style="position:absolute;top:219;left:360"><nobr><span class="fc1">, Alan Kwan</span></nobr></DIV>
<DIV style="position:absolute;top:216;left:450"><nobr><span class="fc2">1</span></nobr></DIV>
<DIV style="position:absolute;top:219;left:455"><nobr><span class="fc1"> and Krithi Ramamritham</span></nobr></DIV>
<DIV style="position:absolute;top:216;left:636"><nobr><span class="fc2">2</span></nobr></DIV>
<DIV style="position:absolute;top:219;left:642"><nobr><span class="fc1"> </span></nobr></DIV>
<DIV style="position:absolute;top:249;left:149"><nobr><span class="fc1">Department of Computer Science</span></nobr></DIV>
<DIV style="position:absolute;top:246;left:384"><nobr><span class="fc2">1</span></nobr></DIV>
<DIV style="position:absolute;top:249;left:390"><nobr><span class="fc1"> </span></nobr></DIV>
<DIV style="position:absolute;top:270;left:163"><nobr><span class="fc1">City University of Hong Kong </span></nobr></DIV>
<DIV style="position:absolute;top:290;left:115"><nobr><span class="fc1">83 Tat Chee Avenue, Kowloon, Hong Kong </span></nobr></DIV>
<DIV style="position:absolute;top:249;left:487"><nobr><span class="fc1">Department of Computer Science and </span></nobr></DIV>
<DIV style="position:absolute;top:270;left:574"><nobr><span class="fc1">Engineering</span></nobr></DIV>
<DIV style="position:absolute;top:267;left:659"><nobr><span class="fc2">2</span></nobr></DIV>
<DIV style="position:absolute;top:270;left:665"><nobr><span class="fc1"> </span></nobr></DIV>
<DIV style="position:absolute;top:290;left:481"><nobr><span class="fc1">Indian Institute of Technology Bombay </span></nobr></DIV>
<DIV style="position:absolute;top:310;left:539"><nobr><span class="fc1">Mumbai, India 400076  </span></nobr></DIV>
<DIV style="position:absolute;top:330;left:163"><nobr><span class="fc1">Email: cskylam@cityu.edu.hk </span></nobr></DIV>
<DIV style="position:absolute;top:330;left:536"><nobr><span class="fc1">Email: krithi@iitb.ac.in </span></nobr></DIV>
<DIV style="position:absolute;top:350;left:118"><nobr><span class="fc3"><b> </b></span></nobr></DIV>
<DIV style="position:absolute;top:381;left:230"><nobr><span class="fc3"><b>Abstract </b></span></nobr></DIV>
<DIV style="position:absolute;top:410;left:113"><nobr><span class="fc11">RTMonitor is a real-time data management system <br>for  traffic navigation applications. In
our system, <br>mobile vehicles initiate time-constrained <br>navigation requests and RTMonitor calculates and <br>communicates the best paths for the clients based
<br>on the road network and real-time traffic data. The <br>correctness of the suggested routes highly depends <br>on how well the system can maintain temporal
<br>consistency of the traffic data. To minimize the <br>overheads of maintaining the real-time data, <br>RTMonitor adopts a cooperative and distributed <br>approach
using mobile agents which can greatly <br>reduce the amount of communications and <br>improves the scalability of the system. To <br>minimize the space and message
overheads, we <br>have designed a two-level traffic graph scheme to <br>organize the real-time traffic data to support <br>navigation requests. In the framework, the
agents <br>use an Adaptive PUSH OR PULL (APoP) scheme <br>to maintain the temporal consistency of the traffic <br>data. Our experiments using synthetic traffic data
<br>show that RTMonitor can provide efficient support <br>to serve navigation requests in a timely fashion. <br>Although several agents may be needed to serve a
<br>request, the size of each agent is very small (only a <br>few kilobytes) and the resulting communication <br>and processing overheads for data monitoring can
<br>be maintained within a reasonable level. </span></nobr></DIV>
```

Fig. 2. Partial Content of newfile-1.html

4 HMM Definition and Parameters Estimation

HMM comprises of two sequences, i.e., state sequence and observation sequence. In the PDF metadata extraction, states correspond to the types of text blocks, observation symbols correspond to the content of text blocks. Metadata Extraction is performed by determining state sequence that was most likely to generate the observation sequence.

The header information of each literature includes title, author, affiliation, address, email, abstract, etc. Thus, the state set of HMM is defined as: X = {title, author,

affiliation, address, email, abstract, **final**}. If a text block is the last one in the literature, then we define that the next state of current state is the **final** state. The observation symbols contain all the words and numbers in all text blocks, which can be obtained through statistics of all training literatures. The initial state distribution contains the probabilities of each state that is in the first text block. The observation probability distribution contains the probabilities of each word appeared in each state.

In this paper, we have a text block as the smallest unit of information processing. Each text block corresponds to a unique state. Traditionally, HMM contains only one state transition matrix, which treats the whole literature as a data stream of text blocks. Although this approach retains the text blocks sequential relationship, it loses the text blocks location relationship. However, the location relationship is very important to determine the type of text block. For example, if we know a text block indicates the author, then the text blocks on the same line is very likely to indicate author too. In Fig. 2, the text blocks from the 5^{th} to the 11^{th} row are all authors, and they are on the same line. Of course, the punctuation, number and other symbols are not author names, but they are ancillary information that associated with the authors. For example, the superscript 1 denotes that the author in front of it belongs to the first affiliation.

Through the above analysis, we can draw the following conclusions: 1) in the same line, the transition probability between the same states is **much larger** than the normal one; 2) in the same line, the transition probability between different states is **smaller** than the normal one; 3) in different lines, the transition probability between the same states is **smaller** than the normal one; 4) in different lines, the transition probability between different states is **larger** than the normal one. These conclusions have been verified through the experiment results in Table 1, Table 2 and Table 3. The original state transition matrix A has been divided into two matrixes, A' A''.

Table 1. The Original State Transition Probability Matrix A

Current \ Next	title	author	affiliation	address	email	abstract	final
title	**0.3181**	0.6750	0.0000	0.0000	0.0000	0.0068	0.0000
author	0.0000	**0.4095**	0.5521	0.0000	0.0356	0.0026	0.0000
affiliation	0.0000	0.0146	**0.3532**	0.2775	0.3067	0.0464	0.0013
address	0.0000	0.0000	0.0092	**0.3588**	0.5797	0.0521	0.0000
email	0.0000	0.2528	0.1245	0.0000	**0.1566**	0.4452	0.0207
abstract	0.0026	0.0066	0.0000	0.0000	0.0000	**0.6071**	0.3834

Table 2. The State Transition Probability Matrix in the Same Line A'

Current \ Next	title	author	affiliation	address	email	abstract	final
title	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
author	0.0000	**0.9903**	0.0048	0.0000	0.0000	0.0048	0.0000
affiliation	0.0000	0.0000	**0.8783**	0.0684	0.0532	0.0000	0.0000
address	0.0000	0.0000	0.0731	**0.8536**	0.0731	0.0000	0.0000
email	0.0000	0.0000	0.0000	0.0138	**0.9861**	0.0000	0.0000
abstract	0.0000	0.0000	0.0000	0.0000	0.0000	**1.0000**	0.0000

Table 3. The State Transition Probability Matrix in Different Lines A''

Current \ Next	title	author	affiliation	address	email	abstract	final
title	**0. 3055**	0.6944	0.0000	0.0000	0.0000	0.0000	0.0000
author	0.0000	**0.0186**	0.9285	0.0000	0.0527	0.0000	0.0000
affiliation	0.0000	0.0048	**0.2396**	0.3374	0.3300	0.0855	0.0024
address	0.0000	0.0000	0.0139	**0.3162**	0.5860	0.0837	0.0000
email	0.0000	0.0526	0.1085	0.0000	**0.0296**	0.7894	0.0197
abstract	0. 0000	0.0027	0.0000	0.0000	0.0000	**0.5931**	0.4041

Now that whether is in the same line has great impact for the transition probability between states, how can we know whether the two text blocks are in the same line? We proposed a simple way to solve it. Firstly, all text blocks are aggregated into multiple groups. In each group, the difference between the max **top** value and the min **top** value should be less than a threshold. This approach is basically correct. The recommended threshold value can be 6-10.

5 Viterbi Algorithm Improvement and Enhancement

Viterbi algorithm uses dynamic programming technology. Firstly, we briefly introduce how Viterbi algorithm finds the "best" state sequence.

We define $\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} P[q_1 q_2 ... q_{t-1}, q_t = i, O_1, O_2, ... O_t \mid \lambda]$

What we try to find is the state sequence that corresponds to the maximum $\delta_t(i)$ at time t. The steps are as follows:

1) Initialization

$\delta_t(i) = \pi_i b_i(O_1),\ 1 \le i \le N; \quad \phi_1(i) = 0,\ 1 \le i \le N;$ (1)

2) Recursive

$$\delta_t(j) = \max_{1 \le i \le n}[\delta_{t-1}(i) a_{ij}] b_j(O_t),\ 2 \le t \le T;\ 1 \le j \le N; \quad (2)$$

$$\phi_t(j) = \arg\max_{1 \le i \le n}[\delta_{t-1}(i) a_{ij}],\ 2 \le t \le T;\ 1 \le j \le N; \quad (3)$$

3) Terminate

$$P^* = \max_{1 \le i \le N}[\delta_t(i)]; \qquad q_t^* = \arg\max_{1 \le i \le N}[\delta_t(i)] \quad (4)$$

4) Find the S sequence

$$q_t^* = \phi_{t+1}(q^*_{t+1}), \qquad t = T-1, T-2 \ldots, 1 \quad (5)$$

We can improve the Viterbi algorithm by adjusting two factors, i.e., B and A. In this paper, the emission probability of each text block is the sum of that for all words in the text block. Assume that the block sequence is: $O = O_1 O_2 \ldots O_T$, and the length of the t^{th} block is K (i.e., it contains K words), $O_t = O_{t1}\ O_{t2}\ \ldots O_{tK}$. The probability that state j emits the t^{th} block is:

$$b_j(O_t) = \sum_{k=1}^{K} b_j(O_{tk}) \tag{6}$$

We found that some $b_j(O_t) = 0$, which lead to the extraction accuracy decrease remarkably. In order to avoid this problem, we assign a very small value for them (about one ten-thousandth), and the accuracy return normal level. Similarly, for the transition probability $a_{ij} = 0$, we also assign a very small value to them (about one thousandth).

In the formula (2) and (3), a_{ij} denotes the transition probability from state i to state j, which did not concern the location information of text blocks. Based on the discussion in section 4, we can modify the formula (2) as follows:

$$\delta_t(j) = \begin{cases} \max\limits_{1\le i\le n}[\delta_{t-1}(i)a'_{ij}]b_j(O_t) & line(O_t) = line(O_{t-1}) \ \ 2\le t\le T \\ \max\limits_{1\le i\le n}[\delta_{t-1}(i)a''_{ij}]b_j(O_t) & line(O_t) \ne line(O_{t-1}) \ \ 1\le j\le N \end{cases} \tag{7}$$

The function **line** returns the line number of the text block. From the formula (7) we can see that, if the two adjacent text blocks belong to the same line/group, then the matrix A' should be used; otherwise, the matrix A'' should be used.

6 Experiments and Analysis

Because there are no text blocks location information in the existing dataset, we downloaded and compiled more than 500 PDF literatures from VLDB conferences manually. However, some literatures became garbled after treatment by the tool pdftohtml-0.39. At last, we collected 458 valid PDF literatures. These literatures are divided into two groups: the former 300 as the training dataset, and the latter 158 as the test dataset. We evaluate the algorithm performance using the precision and recall, which are defined as follows:

Precision for each state is defined as the number of text blocks with the correct tag for the state divided by the total number of text blocks with the machine tag for the state.

Recall is defined as the number of text blocks with the correct tag for the state divided by the total number of text blocks with the manual tag for the state.

We did a total of three experiments. In the first experiment, we use the text block based way to extract metadata. The precision and recall are shown in the 2nd and 3rd columns of Table 4. The text block based way is superior to the word based way, which has been verified in reference [7], so we did not do that experiment again. In the second experiment, we considered the text block location information, and computed the new state transition matrixes as shown in Table 2 and Table 3. By using the two matrixes and the formula (7), the accuracy of Viterbi algorithm on average is increased about 4.5%. We can see that title precision and recall, author precision and recall, affiliation precision and recall, address precision, email recall and abstract precision are all increased remarkably.

Table 4. The Precision and Recall for Text Block Location Based Way

Accuracy State	Text Block Based Way		Location Based Way	
	Precision	Recall	Precision	Recall
title	0.975207	0.925490	**1.000000**	**1.000000**
author	0.894488	0.948247	**0.993266**	**0.984975**
affiliation	0.869469	0.825630	**0.930526**	**0.928571**
address	0.738739	0.858639	**0.943182**	0.869110
email	0.933993	0.792717	0.941333	**0.988796**
abstract	0.961538	0.992063	**0.990777**	0.994709

7 Related Works

Existing literature metadata extraction can be divided into two categories: extract metadata from the paper list on a Web page; extract metadata from a lot of PDF literatures. They are usually concerned with different metadata items. The former mainly extract the title, author, journal, year, conference, column, issue, page, etc; the latter mainly extract the title, author, affiliation, address, email, abstract, keyword, etc. Both of them can be extracted using HMM. Most HMMs are constructed automatically [8, 9]. For example, Seymore [9] proposed to extract the title, author, keyword and other metadata for each literature using HMM in Cora [4], which has an average accuracy of up to 92.9%.

In order to improve the accuracy of HMM-based metadata extraction algorithm, many people attempted to take full advantage of the feature of the text. Ling Zhang [10] proposed a HMM structure learning method based on symbol feature extraction, in which each feature corresponds to a state. She also modified Viterbi algorithm through add a type parameter to each state. The extraction accuracy of her algorithm is superior to that of [9]. Ming Zhang et al. [11] proposed a hybrid statistic model for metadata extraction: SVM + BiHMM. The BiHMM model modifies the HMM model with both Bigram sequential relation and position information of words, by means of distinguishing the beginning emitting probability from the inner emitting probability. The average extraction accuracy of their model can reach 96.5%.As a result of the different datasets, the accuracy of our algorithm cannot be compared directly to other algorithms. However, the practical application of our algorithm has achieved good results, which has an average accuracy of up to 97.4%.

8 Conclusions

In this paper, we proposed a new PDF academic literatures metadata extraction approach. It used the location information of text blocks and modified the definition of HMM by adding one state transition matrix. The two matrixes reflect such fact that the transition probability between the same states in the same line is far greater than that in different lines. From the above experiments we can see that the accuracy for the improved Viterbi algorithm is increased greatly. We also found that the number or symbol superscript is useful for the metadata analysis. It can tell us which affiliation that each author belongs to. Next, we will be in this area for further research.

Acknowledgments

This Work is Supported by Project of "Taishan Scholar" Funded by Government of Shandong Province and Foundation for Outstanding Young Scientist in Shandong Province.

References

1. Giles, C.L., Kurt, D.B., Steve, L.C.: An automatic citation indexing system. In: Digital Libraries 1998 (1998)
2. Ying, D., Gobinda, C., Schubert, F.: Template mining for the extraction of citation from digital documents. In: Proc. Second Asian Digital Library Conference, Taiwan, pp. 47–62 (1999)
3. Dayne, F., Andrew, K.M.: Information extraction with HMMs and shrinkage. In: AAAI 1999 (1999)
4. Cora Dataset (2003), http://www.cs.umass.edu/~mccallum/data/cora-hmm.tar.gz
5. pdftohtml (2006), http://sourceforge.net/projects/pdftohtml/files/
6. Du, L.: Hidden markov model (HMM), http://math.sjtu.edu.cn/teacher/wuyk/HMM-DL.pdf
7. Cui, B.: Scientific literature metadata extraction based on HMM. In: Luo, Y. (ed.) Cooperative Design, Visualization, and Engineering. LNCS, vol. 5738, pp. 64–68. Springer, Heidelberg (2009)
8. Zhang, N.R.: Hidden markov models for information extraction (June 2001)
9. Seymore, K., McCallum, A., Ronal, R.: Learning hidden markov model structure for information extraction. In: AAAI 1999 Workshop on Machine Learning for Information Extraction (1999)
10. Zhang, L.: Research and application of web information extraction technology. Master's thesis. Chinese Academy of Sciences (2003)
11. Zhang, M., Yin, P., Deng, Z.H., Yang, D.Q.: SVM+BiHMM: A hybrid statistic model for metadata extraction. Journal of Software 19, 358–368 (2008)