Web Information Extraction Using Web-Specific Features

Jinlin Chen
Department of Computer Science
Queens College, City University of New York
jchen@cs.qc.edu

Ping Zhong
Department of Computer Science,
Graduate Center, City University of New York
pzhong@gc.cuny.edu

Journal of Digital Information Management

ABSTRACT: *Several problems exist with traditional HMM based approaches for Web information extraction (IE) due to the lack of consideration on Web-specific features. To address this issue we present a Generalized Hidden Markov Model (GHMM) that extends HMMs by making use of Web-specific information for Web IE. In GHMM based approach, Web content blocks instead of terms are used as basic extraction unit. Besides, instead of using the traditional sequential state transition order, GHMM decides the state transition order based on layout structure of the corresponding web page. Furthermore, GHMM uses multiple emission features derived from Web information instead of single emission feature. Experimental study shows that GHMM based approach can effectively improve Web IE comparing to traditional HMM based approaches.*

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval**]; H.3.5** [Online Information Services]:

General Terms
Web information extraction, Web-specific features

Keywords: Hidden Markov Model, Information extraction

Received 10 Sep. 2007; Reviewed and accepted 27 Jan. 2008

1. Introduction

Today information extraction (IE) has become an important technology for Web information consumption. Most of the current efforts in IE focus on automatic training approach where machine learning methods play a key role. The approach first annotates a training corpus, and then an algorithm is run to train the system for analyzing novel information. There have been two main categories of automatic training approach - the wrapper-based approach [11][13] where rules are automatically discovered using predefined templates, and the statistical generative model where statistical models or parsers are constructed and then decoded to find relevant content. Hidden Markov Models (HMMs) [2][3][5]-[7][12][14], and related probabilistic sequence models [8][9][15], have been among the most accurate methods for automatic training approach.

When applying HMM for Web IE, several problems exist due to the lack of consideration on Web-specific features. First, previous HMM based approaches take single term as basic unit for IE, which is appropriate for pure text information. A Web page usually consists of multiple content blocks in which logically related contents are grouped together. This clustering property of Web content provides additional information for improving Web IE. Second, in order to establish well-structured models, HMMs always follow a specific, stable, and appropriate state transition order. Traditional HMM based approaches assume a sequential state transition order (usually left to right and then top to bottom).). However, it is no longer the case in Web pages — a two-dimensional space. The most stable and appropriate state transition order in Web pages may not necessarily be the left to right and top to bottom order as in traditional pure text. Furthermore, traditional HMM based approaches only consider the semantic term attribute as observed emission feature. Web information contains other emission features such as format, layout, etc, which could be observed to help improve state transition estimation for HMM.

In this paper a Generalized Hidden Markov Model (GHMM) is proposed to extend traditional HMM by making use of Web-specific information for Web IE. First, we use Web content block instead of term as basic extraction unit. Second, instead of using the traditional sequential state transition order for HMM, we detect state transition order based on Web page layout structure. Furthermore, we use multiple emission features (term, layout, and formatting) instead of single emission feature (term). In this way our proposed system can better accommodate Web IE. Experiments show promising results comparing to traditional HMM based Web IE.

The remainder of the paper is structured as follows. Section 2 gives a brief review on previous work. Section 3 explains our motivation. GHMM for Web IE is discussed in section 4. Experiment results are presented in section 5. Conclusions and future work are given in section 6.

2. Previous work

In this section we first give a brief introduction to the concept of HMM and its application on IE. We then give a review on page layout analysis which is relevant to state transition sequence detection.

2.1 HMM and IE

A discrete output, first-order Hidden Markov Model (HMM) is a finite state automaton and can be represented by a 5-tuple $\{S, V, \Pi, A, B\}$, where S is a set of values for the hidden states, $S = \{s_1 \ldots s_N\}$, N is the total number of possible states; V is a set of values for the observations, $V = \{v_1 \ldots v_M\}$, M is the total number of possible observation values; Π is a set of initial probabilities for all the states, $\Pi = \{\pi_i\}$, $i = 1, 2, \ldots, N$; A is a mapping defining the probability of each state transition A: $S \times S \rightarrow [0, 1]$; B is a mapping defining the emission probability of each observation value on each state, B: $S \times V \rightarrow [0, 1]$. The following properness constraints must be satisfied: $\forall s \in S, \sum_{s' \in S} A(s, s') = 1$; $\forall s \in S$, $\sum_{v \in V} B(s, v) = 1$.

Using HMM, we can find out the most probable hidden state sequence corresponding to a given observation sequence. HMMs have been successfully applied to many information extraction tasks such as named entity extraction [2], bibliographic information extraction [14], recovering the sequence of a set of entities occurring in close proximity (dense extraction) [3], as well as the sparse extraction task, in which the object is to extract relevant phrases from documents containing much irrelevant text [5][7][8]. Most of these works took processed data as pure text and aimed at improving parameter and structure estimation. Freitag et al used a statistical technique called shrinkage to improve parameter estimation [5]. Freitag et al also presented an algorithm for automatically finding good structures of HMM by stochastic optimization [7]. McCallm and Lafferty et al [8][9] proposed probabilistic sequence models close to HMMs for IE.

Some other approaches exploit structural information of processed data to improve IE. Skounakis et al [12] used Hierarchical HMM to exploit grammatical structure of sentences instead of single terms for information extraction. Song et al [1] proposed a Mixture HMM approach which makes use of sentence structure to improve IE. Two-dimensional HMM [4] has been proposed for image classification task. However, the 2D-HMM is a general image-based approach which is not suitable for Web IE due to its complexity. Two-dimensional Conditional Random Fields (2D-CRF) [15] approach has been proposed for Web IE. The basic strategy of this approach is to model web content blocks using a 2D grid according to their position and size, and then evaluate the mutual dependencies. Essentially this approach is still image-based and does not make use of Web-specific features.

2.2 Web page layout analysis

Web page layout analysis is important for many applications such as adaptive Web content presentation [10][16], Web information retrieval [21], etc. Many approaches have been proposed for Web page segmentation. In our previous work [10][18] visual clues as well as heuristic based approaches were used for page layout structure detection. Kovacevic et al [22] detected common page areas such as header, footer, left and right menu, and center of a page based on visual information. Cai et al [17] used a vision based page segmentation algorithm to detect hierarchical structure of Web content block. Romero et al [20] partitioned page into content blocks using clustering method directly based on Document Object Model (DOM) [23] tree of Web pages. Vitali et al [19] used a heuristic approach based on DOM tree for analyzing structure of Web pages. Chen et al [16] used a rule based approach based on both visual and DOM information to detect Web page structure.

3. Motivation

The basic idea of our approach is to make use of Web-specific information to improve HMM based IE. Below we give a detailed analysis to these Web-specific features. First, information in a Web page is normally grouped into different regions. Generally speaking, a Web author would organize the content of a Web page for easy browsing and understanding. Thus logically related contents are usually grouped together and the entire page is segmented into content regions using explicit or implicit separators such as lines, blank areas, etc. We call each segmented unit a block. The knowledge of the structure of Web content blocks, as well as other Web-specific features, such as content format and layout which we will address in the following, is believed to help improve Web IE with the observation that block-based HMM structures are usually more reliable and less variable than traditional term-based ones, and extra Web-specific features can complement observation symbols in traditional HMMs.

However, most previous approaches were initially designed for IE on pure text which does not have such information. Correspondingly, these approaches were unable to make use of block information implicitly embedded in Web page design. Two relevant approaches exploit grammatical structure of sentences to improve IE using Hierarchical HMM[12] and Mixture HMM [1]. These approaches are more suitable for traditional text documents. Web information generally lacks the grammatical structure as exploited in these approaches. On the contrary, the clustering property of Web information provides additional clue that content in the same block tends to serve for the same function for most well-designed Web pages and therefore be emitted by the same hidden state. To make use of such additional information, we propose to build HMM based on content blocks instead of single terms in our approach.

Second, a Web page is a two-dimensional space from both design and browsing point of view. The most stable and appropriate state transition order may not necessarily be left to right and top to bottom order as in traditional pure text. Fig. 1(a) shows a Web page with left-right then top-bottom transition sequence, and Fig. 1(b) shows a page with topbottom then left-right transition sequence. If we use top-bottom then left-right transition sequence for (a) and use left right then top-bottom block transition sequence for (b), as shown in Fig. 1 (c) and (d), respectively, then the constructed HMMs will be inappropriate for IE.

To find out an appropriate state transition order, observing that visual layout structure of a Web page reveals the logical relationship among different blocks, in this paper we propose a layout structure based state transition sequence detection method. The basic idea is that this state transition sequence, in which page elements following a logical relationship, is rather stable and appropriate. In next section we will give the detail of our approach.

2D-CRF [15] also attempts to capture 2-dimensional neighborhood interactions for Web IE. However, this approach

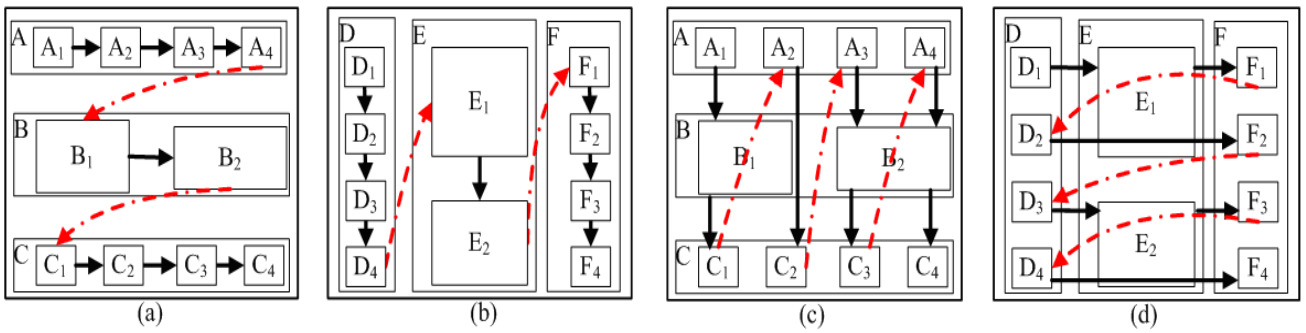


Figure 1. Example of different transition sequences

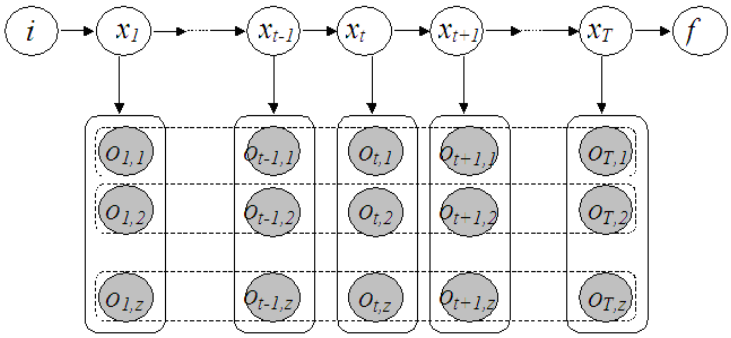


Figure 2. Generalized Hidden Markov Model

assumes a uniformed neighborhood interaction and fails to observe that state transition still follows certain sequence as revealed by hierarchical layout structure.

Furthermore, traditional HMM based approaches only consider the semantic terms as observation symbols. Web information contains additional observable information such as content formatting information and layout information. Combined with the semantic feature of the terms, these extra attributes provide more knowledge about Web information, which we believe can improve Web IE.

To estimate the impact of the above mentioned observations, we made a statistical analysis by randomly selecting 20 Web IE tasks from various Web pages, such as air ticket prices, course titles, NBA players rebounds statistics, desktop deals and offers, MP3 player models, etc. Our preliminary analysis reveals that almost 85% and 90% of the tasks would benefit from the block-based Web IE and multiple attributes incorporation, respectively. Based on this analysis, we propose a Generalized Hidden Markov Model to improve Web IE.

4. Generalized Hidden Markov Model for Web IE

4.1 GHMM

Fig. 2 shows the GHMM model which starts from the initial state *i*, transits to x_1, emits symbol *O*1 with attributes 1,2,…,z, then transits to x_2, emits symbol *O*2 with attributes 1,2,…,z, and so on until it reaches the final state *f*.

Table 1 defines related notations of GHMM, where *N* is number of states in the model, *Z* is the number of the attribute sets of observation symbols, *Ms* is the number of distinct observation symbols w.r.t. the attribute *s*,$1 \le s \le Z$, and *T* is the number of symbols observed.

Goal definition: Given the model $\lambda = (A, B, \Pi)$ as defined in Table 1, find out the state transition sequence $I = i_1, i_2, \ldots, i_T$ which maximizes $P(O, I \mid \lambda)$.

$$P(O, I \mid \lambda) = P(O, I \mid \lambda)\, P(O, I \mid \lambda) = \pi i_1\, b\, i_1\, (o_1) a_{i}, i_2\, b\, i_2\, (o_2) \ldots\ldots a i_{T-1} i_T\, b\, i_r\, (o_T) \quad (1)$$

If we extend an observation symbol *k* from its mere term attribute to attributes as *k1*, *k2*, …, *kz* , various attributes of the observation symbol can be included. We assume that the attributes are independent from each other and consider a linear combination of them, then we have

$$b_j (k \sum_{s=1}^{z} \alpha \quad s * b_j (k_s)$$

where αs is the weight factor for the s^{th} attribute, $\sum_{s=1}^{z} \alpha\ s = 1,0 \le \alpha s \le 1, 1 \le s \le z, b_j, (k_s)$ is the probability of observing the attribute *ks* of symbol *k* given that we are in state *j*.

Equation (1) now becomes

$$P(O, I \lambda)) = \pi_{i1} \left[\sum_{s=1}^{z} \alpha\ s^* \overset{s}{bi}_1(o_1,s)\right] a_{i1},i_2 \left[\sum_{s=1}^{z} \alpha\ s^* \overset{s}{bi}_2(o_2,s)\right]$$

$$\ldots\ a i_T\text{-}1^{i}{}_{T} \left[\sum_{s=1}^{z} \alpha\ s^* \overset{s}{bi}_T(o_T,s)\right] \quad (2)$$

The extended Viterbi algorithm inductively keeps the most probable state sequence for each of the *N* states as the intermediate state at each instant for the desired observation sequence. Note that here each observation of emission consists of multiple attributes of symbols. To calculate the probability of a partial observation sequence at a given state and instant, it requires O(*ZN*) other than O(*N*) because of the multiple attributes of observation symbols. Since it needs to compute the probability for *T* time instants and *N* states, it is easy to find that the overall time complexity is O(*ZN*2*T*).

$V = \{V^1, V^2, \ldots, V^Z\}$,where $Vs = \{ v1, v2, \ldots, v_{Ms}\}$ and $1 \le s \le Z$	V^s, the distinct set of possible observation symbols w.r.t. the attribute *s*
$\prod = \{\pi 1, \pi 2, \ldots, \pi N\}$, where $\pi i = P\{i_1 = i\}, 1 \le i \le N$	πi, the probability of being in state *i* at the beginning of the experiment, i.e.at *t*=1
$A = \{ aij \}$ and $aij = P\{i_{t+1} = j \mid i_t = i\}$, $1 \le i, j \le N$	Transition probability a_{ij}, the probability of being in state *j* at time *t+1* given that we are in state *i* at time *t*
$B = \{B^1, B^2, \ldots, B^Z\}$,where $B^s = \{b_j(k_s)\}$, $b_j(k_s) = P\{v_k$ at $t \mid i_t = j\}$, $1 \le s \le Z$, $1 \le k_s \le M_s$, $1 \le j \le N$	Emission probability $b_j(k_s)$, the probability of observing the symbol v^s_k w.r.t. the attribute *s* given that we are in state *j*
$\lambda = (A, B, \Pi)$	Notation of generalized HMM

Table 1. Notation of GHMMs

In (2), if we set $\alpha_s = 1$ for $s = 1$, and $\alpha_s = 0$ for $2 \le s \le z$, that is, we choose only one observation symbol set (the term attribute in this case), we can reduce (2) to (1). In this case GHMM is reduced to the traditional HMM.

4.2 Generalization

In our GHMM, two major generalizations are made. (1) In traditional HMMs, each word in the training data is assigned to its own state, which transits to the state of the word that follows it. Our proposed GHMMs conduct similar process except that we construct the models on the basis of blocks. That is, each block is assigned to its own state, a block transits to the state of the block that follows it. We believe that for Web pages block-based HMMs will outperform term-based HMMs since block-based HMM structures are usually more reliable and less variable than term-based HMM structures. In a Web page, the number of blocks is much less than that of terms, thus spatial arrangement of blocks within a page is obviously less complicated and more fixed than that of terms. In addition, a Web author tends to apply the same design style to many pages on the block basis, but not necessarily follows that on the term basis.

Fig. 3 and Fig. 4 show examples of term-based and block-based HMMs with certain transition sequence. Marking on Fig. 3(a) is incomplete yet for the purpose of simplicity, but it is sufficient to demonstrate the idea. Fig. 4 represents terms by circles and blocks by rectangles. To make sure that terms inside the same block are emitted by the same hidden states, currently we limit our block to basic Web elements as well as groups of basic elements within a larger block that share the same presentation scheme. In (X)HTML basic elements are elements created by a pair of (X)HTML tags without further tags inside. (*Note*: if a block contains irrelevant information, i.e., multiple states exist within the same block, we can borrow the idea of hierarchical Hidden Markov Models [12] and apply the model to first extract the block and then its subdivisions. We leave this as part of our future work.)

To detect content blocks in a Web page, we use a visual based approach that we have previously presented in [10] and [18].

(2) Traditional HMMs only consider the term attribute as observation symbols. Web information contains additional observable information that can be exploited. Currently we only select three attributes for each Web block: term, format, and layout, which can and should be extended to other Web-specific features such as link structure. Term attribute concerns the semantic terms similar to traditional HMMs. Format attribute concerns terms' text and font attributes which can be obtained from Cascading Style Sheet (CSS). Layout attribute concerns how subdivisions within a block are displayed or organized. In our model, we choose several typical CSS classes of the Web pages as the format attribute of observations; we roughly categorize some basic layouts and choose them as the layout attribute of observations.

Note: in our model we assume that every block only contains desired information, which is not always the case because of irrelevant noises. We also simplify dependency relationship among different observation symbol attributes. For instance, Web pages from the same Web site tend to have a similar layout given the same format. Their dependencies are rather complicated and we leave them for future study. We implement GHMM based on the proposed algorithm. Here we denote an observation symbol k with attributes as $k1$, $k2$, $k3$ which correspond to term, format, and layout attribute, respectively. As mentioned above, we consider the emission probability as a linear combination of these attributes for simplification purpose, that is,

$$b_j(k) = \alpha_1 * b_j(k_1) + \alpha_2 * b_j(k_2) + \alpha_3 * b_j(k_3)$$

where α_1, α_2, and α_3 are the weights for term attribute, format attribute, and layout attribute respectively, and $\alpha_1 + \alpha_2 + \alpha_3 = 1$, $0 \le i \le 1$. Here different weight combinations may have different impact on web IE. The optimal weight set may vary on different tasks. In section 5 we will further examine this issue with experiment.

4.3 State transition sequence detection

State transition sequence of a typical Web page may be quite different from that of a pure text page due to the 2- dimensional

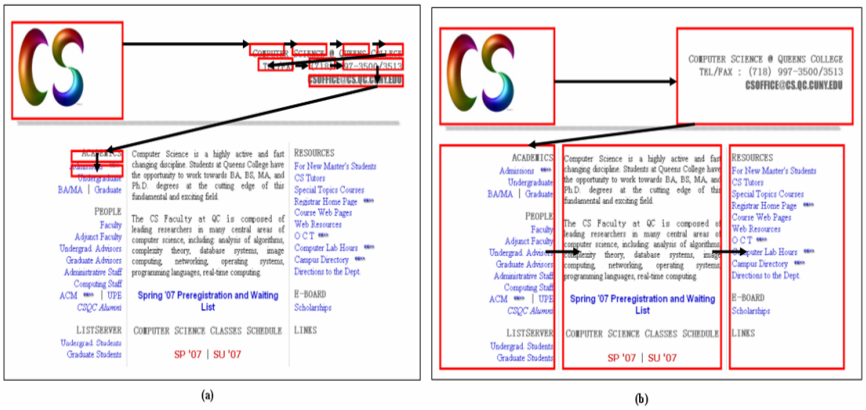


Figure 3. Web page example of term-based vs. block-based transition (a) Term-based (b) Block-based

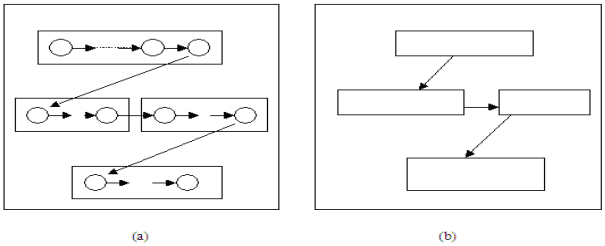


Figure 4. Term-based vs. Block-based transition (a) Term-based (b) Block-based

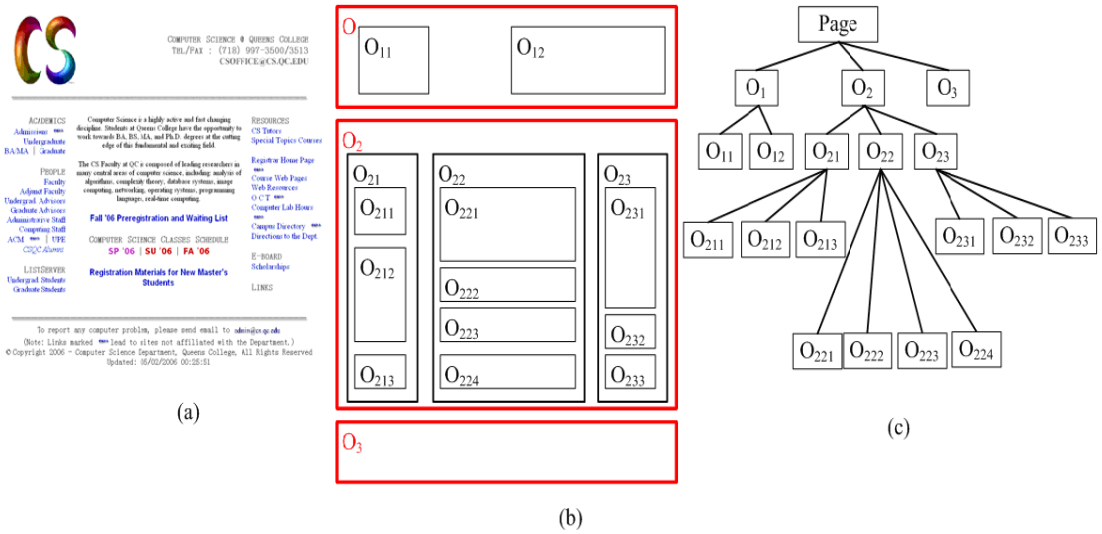


Figure 5. Layout structure of a Web page (a) Original page (b) Layout structure (c) Abstract layout structure

nature of Web pages from both design and browsing point of view. Since the hierarchical layout structure, based on which different elements of the page can be grouped in an order conforming to their logical relationship, provides an important clue for state transition sequence due to their common nature originated from the authoring process and with high possibility retains throughout all pages, we consider the resulting state transition sequence as an ideal choice applied to our model.

For example, for the Web page shown in Fig. 5(a), the corresponding layout structure is shown in Fig. 5(b), which can be further abstracted using a hierarchical structure as shown in Fig. 5(c). Traversing this abstract hierarchical layout structure of the Web page using a depth-first traversal algorithm (either pre-order or post-order), and discarding non leaf nodes in the traversal result, the resulted sequence of Web elements reveals the state transition sequence of the GHMM corresponding to the page. The rationale is that logically this traversal sequence can be considered as the most stable and appropriate sequence that page elements follow throughout all pages. Our experiments demonstrate the effectiveness of such sequence.

For the example page shown in Fig. 5(a), the corresponding transition sequence should be O_{11}, O_{12}, O_{211}, O_{212}, O_{213}, O_{221}, O_{222}, O_{223}, O_{224}, O_{231}, O_{231}, O_{233}, and O_3. To detect the layout structure of a Web page, we use our previous approach based on visual clue [10][18]. This approach can detect the hierarchical structure of layout to the basic building block level of a Web page (which is a basic HTML element) and abstract all possible types of logical relationship among Web blocks.

4.4 Web IE using GHMM

The system framework of Web IE using GHMM is shown as Fig. 6, which consists of training and testing stage. At training stage, we first perform layout detection (step 1 in Fig. 6) on the training Web pages to segment each ofthem into blocks and discover the hierarchical layout structure as the state transition sequence for the model; at the same time, we conduct multiple attribute analysis (step 2) on each page to obtain desired Web-specific attributes information. We then train the GHMM (step 3) based on the above layout structure, block information, attribute information, as well as manually labeled state information. A GHMM for corresponding IE task is the output of training stage.

At testing stage, we first perform similar layout detection (step 4) and multiple attribute analysis (step 5) on the testing pages. We then apply all the resulting information to the trained GHMM (step 6) from the previous stage to analyze the optimal state of every block in a page. Eventually labeled information of the states of each block will be created and serve as guideline for Web IE.

5. Experiments

5.1 Testing environment

Several popular testing data sets for IE such as ISI RISE (http://www.isi.edu/info-agents/RISE/), Pascal Challenge (http://nlp.shef.ac.uk/pascal/), and Information Extraction Corpus (www.grappa.univ-lille3.fr/~marty/corpus.html) exist. However, most of these data sets are more pure text information oriented.

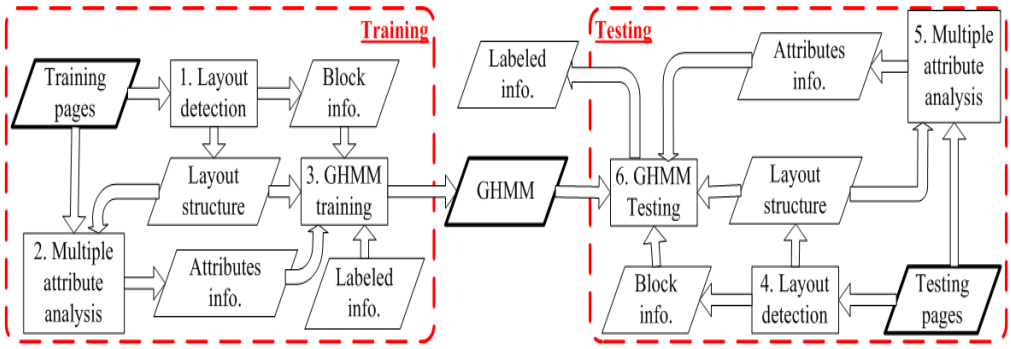


Figure 6. Framework of Web IE using GHMM

Even though some of them are coded in (X)HTML format, generally speaking the layout schemes are rather simple, and most of the transition sequences are left to right then top to bottom, which is quite similar to pure text information. Because of this, in this paper, in addition to use some data from these traditional testing data sets, we also use Web pages from popular websites as testing data.

We have made these new testing data available online in our website at www.alpha.cs.qc.edu/iedata for comparison purpose.Three types of experiments were performed to verify the effectiveness of the strategies we proposed for GHMM based Web IE, i.e., block-based GHMM, multiple-attributes incorporation, and layout structure based state transition sequence for GHMM.

We run our experiments on a Dell Optiplex desktop with 3.2 GHz Pentium 4 processor, 1GB main memory, and Ultra-ATA 80GB hard disk. The operating system is Windows XP.

5.2 Block-based vs Term-based HMMs

To compare term-based HMMs and block-based HMMs we first design an experiment of extracting information from the headers of research papers. The header of a research paper consists of words preceding the main body of the paper, and includes the title, author names, emails, addresses and keywords. We first search on www.google.com using the keywords *information extraction* and *pdf*, and then click the link "View as HTML" to obtain a total of 81 research papers as training pages and two sets of testing pages with 47 and 38 papers, respectively. The experiment results are shown in Figure 7. The average precision of each approach is shown as Table 2.

	Exp. (a)	Exp. (b)
Block-based HMMs	0.900	0.895
Term-based HMMs	0.832	0.828

Table 2. Average precision of term-based and block-based HMMs

Although these pages are not representative HTML and contain natural Web blocks, they are sufficient to demonstrate that block-based HMMs perform much better compared to term-based HMMs in Fig. 7 and Table 2. This conforms to our analysis in previous sections.

	IAF - alt.name	*IAF - organization*
GHMM	30.8	65.5
HMM	1.7	16.8
Stalker	100	48.0
BWI	90.9	77.5

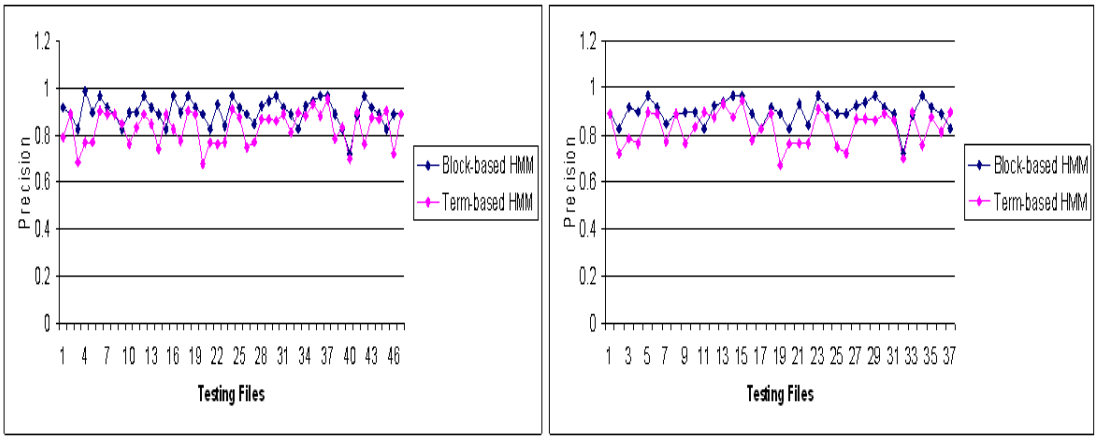
Table 3. Precision Comparison of Different Approaches for IAF

To further examine the effectiveness of block-based approach, we also evaluate our method on the task of Internet Address Finder (IAF), which is a commonly used task in IE community. The corresponding dataset is available online at http://www.isi.edu/info-agents/RISE/repository.html. The extracted fields include IAF-*altname* and IAF-*org*. Table 3 shows the results of our approach as well as those obtained by some other string-based methods: an algorithm based on Hidden Markov Models (HMMs) [5], the Stalker wrapper induction algorithm [13] and BWI [11].

Since IAF has relatively fixed and structured data formats and sparse corpse, they are more suitable for extraction using wrapper induction methods. This is why Stalker and BWI perform well in this task. Our proposed GHMM, however, is still comparable to Stalker and BWI in terms of extraction precision, and clearly outperforms pure-text term-based HMM.

5.3 GHMMs Incorporating Multiple Attributes

To investigate how multiple attributes of observation symbols under GHMMs contribute to Web information extraction, we design a task of extracting functional blocks on Web pages from www.dell.com. The size of dataset is not significant here since we only attempt to discover the impact of multiple attributes. We use 30 training pages, 10 and 8 testing pages for experiment (a) and (b), respectively. In this experiment, we have 16 functions as states which are *pre_header, header, navigation_list, customer_action, back_to, main_title_image, main_introduction, main_details, main_list, related, note, copyright, textsize, seperator, product,* and *print*. Among the three emission attributes, the term attribute simply corresponds to the words within blocks. To simplify our implementation, we only choose 6 major format classes from the CSS files, which are *para, segmentertile_seg, segmenter_right, lnk, para_small, lnk_iconic* and 5 basic layouts shown as Fig. 8. Although they are enough to represent the layouts of blocks in our experiment, there exist many other layouts which we can add into our consideration in the future.



(a) (b)

Figure 7. Term-based vs. Block based HMM

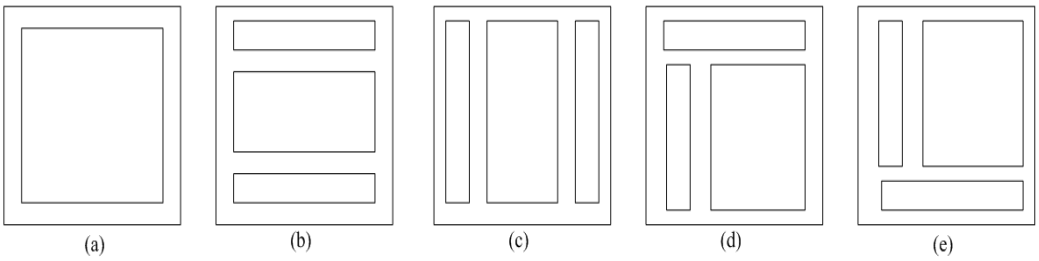


Figure 8. Five basic layout forms

Effects of different weight factor sets of α_1, α_2, and α_3 on extraction results are studied. We list 11 sets of weight factors α_1, α_2, and α_3 in Table 4. The selection of weight factor sets is based on the typical changing points during the experimentation. Note that Set 1, 2, 3 correspond to block-based HMMs considering only term attribute, format attribute, and layout attribute, respectively.

Weight Factor Set No.	1	2	3
1	1	0	0
2	0	1	0
3	0	0	1
4	0.7	0.2	0.1
5	0.5	0.2	0.3
6	0.3	0.3	0.4
7	0.2	0.4	0.4
8	0.2	0.3	0.5
9	0.1	0.4	0.5
10	0.2	0.2	0.6
11	0.1	0.1	0.8

Table 4. Different Weight Factors For Attribute Set

The experiment results of Web IE using GHMMs with different weight factor sets are shown in Figure 9 and Table 5. From Figure 9 and Table 5, we can find that Set 3 outperforms Set 1, which means that the extraction result is better if we only consider the layout attribute rather than the original term attribute. It turns out that careful selection of the attributes of observation symbols can improve Web IE.

In addition, we can find that most set options such as Set 6, 7, 8, and 9, outperform Set 1, which means that appropriate combination of multiple attributes can improve Web information extraction. Particularly, we observe that Set 6 and 7 always give the best extraction results. Our further experiments show that in this case Web IE performs best when $\alpha_1 + \alpha_2 = 0.6$ and $\alpha_3 = 0.4$.

Obviously, many other attributes of Web information such as link structure, etc., can be and should be considered in the future. An important issue is to discover the optimal attribute weight factor sets.

	Exp. (a)	Exp. (b)
Set 1, 2, 5	0.662	0.659
Set 3,	0.729	0.704
Set 4	0.692	0.659
Set 6, 7	0.786	0.883
Set 8, 9	0.786	0.767
Set 10, 11	0.729	0.717

Table 5. Average precision of different GHMMs

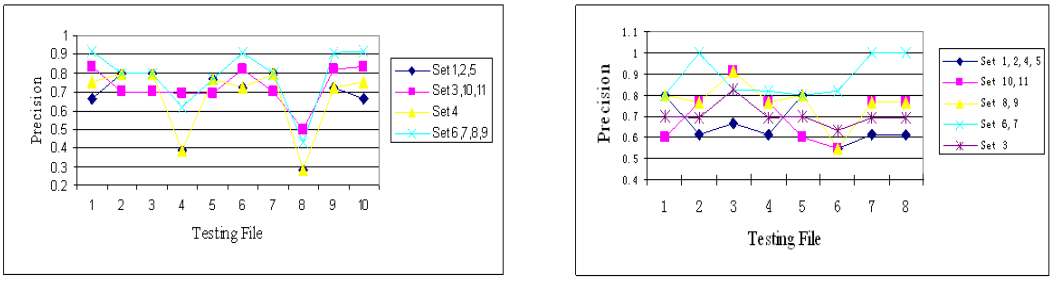
5.4 GHMM with State Transition Sequence based on Layout Structure

In this experiment we verify the effectiveness of our proposed state transition sequence for GHMM based Web IE. Here the task is the same as defined in previous experiment, i.e., to extract functional block from Web pages at www.dell.com. Among a total of 75 pages, we use 50 as training and the rest 25 as testing.

Three different approaches were applied for the testing data: (1) GHMM with the proposed state transition sequence; (2) GHMM with left-to-right and then top-to-bottom state transition sequence; (3) single-emission HMM (Here we only consider term as emission feature).

Approach	Ave. Accuracy
GHMM with the proposed transition sequence	85.5%
GHMM with left-right and top- bottom sequence	79.6%
Single-emission HMM	68.9

Table 6. Comparison of Average precision for extracting functional blocks from pages of www.dell.com



(a) (b)

Figure 9. GHMMs with Multiple Attributes under Different Weight factor Sets

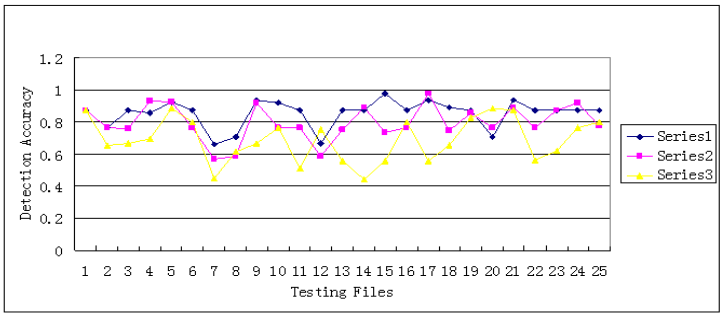


Figure 10. Function detection precision using different models on Web pages from www.dell.com

The average function detection precision of each approach is shown as Table 6. Detailed detection precision for each testing file is shown as Fig. 10, in which *Series* 1, 2, and 3 correspond to approaches (1), (2), and (3), respectively. Although effects of different weight factor sets of α 1, α 2, and α 3 values on detection accuracy results are studied, we only present the result with the optimal weight factor set in this task, in which α 1 = 0.3, α 2 = 0.3, and α 3 = 0.4.

From Table 6 and Fig. 10 we can find that GHMM with the state transition sequence based on layout structure outperforms GHMM with left-to-right and then top-to-bottom transition sequence. Both of them outperform term-based HMM. This verifies the effectiveness of GHMM based Web IE using the proposed state transition sequence and multiple emission features.

6. Conclusions

Our proposed GHMMs utilize segmented units in a Web page called blocks and multiple attributes to extract Web information. Besides, instead of using the sequential state transition order applied in traditional HMM based IE, in our approach we decide state transition order based on the hierarchical layout structure of Web pages, which better accommodate the specific features of Web information. Experiments verify the effectiveness of our proposed strategies and show that GHMMs outperform normal HMMs for Web IE due to the utilization of Web-specific features.

References

[1] Song, M., Song, I.Y., Hu, X., Allen, R. B. (2005). Integrating Text Chunking with Mixture Hidden Markov Models for Effective Biomedical Information Extraction. Proceedings of Computational Science – ICCS 2005. 976-984.

[2] Bikel, D. M., Miller, S., Schwartz, R., Weischedel, R. (1997). Nymble: a high-performance learning namefinder. Proc. of 5th Conf. on Applied Natural Language Processing. 194-201.

[3] Kristie, S., Andrew, M., Ronald, R. (1999). Learning Hidden Markov Model Structure for information Extraction. AAAI-99 workshop on Machine Learning for information Extraction, AAAI Technical Report WS-99- 11. 37-42.

[4] Li, J., Najmi, A., Gray, R. M. (2000). Image Classification by a Two-dimensional Hidden Markov Model. IEEE Trans on Signal Processing, 48(2), 517-533.

[5] Freitag, D., McCallum, K. A. (1999). Information Extraction with HMMs and Shrinkage. AAAI-99 workshop on Machine Learning for information Extraction, AAAI Technical Report WS-99-11. 31-36.

[6] Leek, R. T. (1997). Information Extraction Using Hidden Markov Model. Master's thesis, USSD.

[7] Freitag, D., McCallum, A. (2000). Information extraction with HMM structures learned by stochastic optimization. Proc. of the Seventeenth National Conference on Artificial Intelligence. 584-589.

[8] McCallum, A., Freitag, D., Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. Proceedings of ICML-00. 591-598.

[9] Lafferty, J., Pereira, F., McCallum, A. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the International Conference on Machine Learning 2001. 282-289.

[10] Chen, J., Zhou, B., Shi, J., Zhang, H., Wu, Q. (2001). Function-based Object Model Towards Website Adaptation. Proc. of WWW-10. 587-596.

[11] Freitag, D., Kushmerick, N. (2000). Boosted wrapper induction. AAAI/IAAI 2000. 577–583.

[12] Skounakis, M., Craven, M., Ray, S. (2003). Hierarchical hidden markov models for information extraction. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence.

[13] Muslea, I., Minton, S., Knoblock, C. (2001). Hierarchical Wrapper Induction for Semistructured Information Sources. Journal of Autonomous Agents and Multi-Agent Systems, 4(1/2), 93-114.

[14] Geng, J., Yang, J. (2004). Automatic extraction and integration of bibliographic information on the Web. Proceedings of Database Engineering and Applications Symposium 2004 (IDEAS '04). 193-04.

[15] Zhu, J., Nie, Z., Wen, J., Zhang, B., Ma, W. (2005). 2D Conditional Random Fields for Web Information Extraction. Proceedings of the 22nd International Conf. on Machine Learning. 1044-1051.

[16] Chen, Y., Ma, W., Zhang, H. (2003). Detecting Web page structure for adaptive viewing on small form factor devices. Proceedings of the 12th international conference on World Wide Web. 225-233.

[17] Cai, D.,Yu, S., Wen, J., Ma, W. (2003). VIPS: a vision based page segmentation algorithm. Microsoft Technical Report, MSR-TR-2003-79.

[18] Gu, X., Chen, J., Ma, W., Chen, G. (2002). Visual Based Content Understanding towards Web Adaptation. 2nd Intl. Conf. on Adaptive Hypermedia and Adaptive Web Based Systems. 164-173.

[19] Vitali, F., Di Iorio, A. Ventura, C. E. (2004). Rule-based Structural Analysis of Web Pages. Proceedings of the VI Workshop DocumentAnalysis Systems (DAS 2004). 425-437.

[20] Romero R., Berger, A. (2004). Automatic Partitioning of Web Pages Using Clustering. Proceedings of the Mobile HCI 2004. 388-393.

[21] Yu, S., Cai, D., Wen, J., Ma, W. (2003). Improving Pseudo-Relevance Feedback in Web Information retrieval Using Web Page Segmentation. Proceedings of Twelfth World Wide Web conference (WWW 2003). 11-18.

[22] Kovacevic, M., Diligenti, M., Gori, M., Milutinovic, V. (2002). Recognition of Common Areas in a Web Page Using Visual Information: a possible application in a page classification. Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02). 250-250.

[23] W3C Document Object Model, http://www.w3.org/DOM/.

[24] Forney, D. (1973). The Viterbi algorithm. Proc. IEEE, 61 (3) 268—278.