Webpage Understanding: an Integrated Approach

Jun Zhu*
Dept. of Comp. Sci. & Tech.
Tsinghua University
Beijing, 100084 China
jjzhunet9@hotmail.com

Zaiqing Nie
Web Search & Mining Group
Microsoft Research Asia
Beijing, 100080 China
znie@microsoft.com

Ji-Rong Wen
Web Search & Mining Group
Microsoft Research Asia
Beijing, 100080 China
jrwen@microsoft.com

Bo Zhang
Dept. of Comp. Sci. & Tech.
Tsinghua University
Beijing, 100084 China
dcszb@tsinghua.edu.cn

Hsiao-Wuen Hon
Web Search & Mining Group
Microsoft Research Asia
Beijing, 100080 China
hon@microsoft.com

ABSTRACT

Recent work has shown the effectiveness of leveraging layout and tag-tree structure for segmenting webpages and labeling HTML elements. However, how to effectively segment and label the text contents inside HTML elements is still an open problem. Since many text contents on a webpage are often text fragments and not strictly grammatical, traditional natural language processing techniques, that typically expect grammatical sentences, are no longer directly applicable. In this paper, we examine how to use layout and tag-tree structure in a principled way to help understand text contents on webpages. We propose to segment and label the page structure and the text content of a webpage in a joint discriminative probabilistic model. In this model, semantic labels of page structure can be leveraged to help text content understanding, and semantic labels of the text phrases can be used in page structure understanding tasks such as data record detection. Thus, integration of both page structure and text content understanding leads to an integrated solution of webpage understanding. Experimental results on research homepage extraction show the feasibility and promise of our approach.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models - Statistical

General Terms

Algorithms, Experimentation

*This work is done when the author is visiting Microsoft Research Asia.

KDD'07, August 12–15, 2007, San Jose, California, USA.

Keywords

Text Processing, Webpage Understanding, Conditional Random Fields

1. INTRODUCTION

The World Wide Web is a vast and rapidly growing repository of information, and various kinds of valuable semantic information are embedded in webpages. Some basic understanding of the semantics of webpages could significantly improve people's browsing and searching experience. For example, in our Windows Live Product Search project (http://products.live.com), we automatically extract structured product information from the web using a template-independent approach to segmenting webpages and labeling HTML elements [29]. By presenting the HTML elements according to their semantic meaning, users could save time from sifting the information from thousands of webpages.

However, little work has been carried out on segmenting and labeling the text contents inside the HTML elements of a webpage. In this paper, we study how to use layout and tag-tree structure in a principled way to help understand text contents on webpages.

1.1 Motivating Example

We have been developing *Libra* (http://libra.msra.cn), an object-level academic search engine, to help scientists and students locate research materials. One of the biggest challenges we are facing in *Libra* is how to understand researchers' homepages better. On a researcher's homepage (see Fig. 1 for an example), two types of information are important. The first one is the contact information of the researcher, such as name, address, email, and phone, and the second one is the academic information including the academic title, affiliation, academic activities, and publications. If we could identify all the related information from every researcher's homepage, *Libra* could become the most comprehensive database about researchers in the world.

However, building a webpage understanding model to identify this information from every researcher's homepage is non-trivial:

1. Webpages are highly heterogeneous. Rule-based webpage understanding methods are no longer applicable.

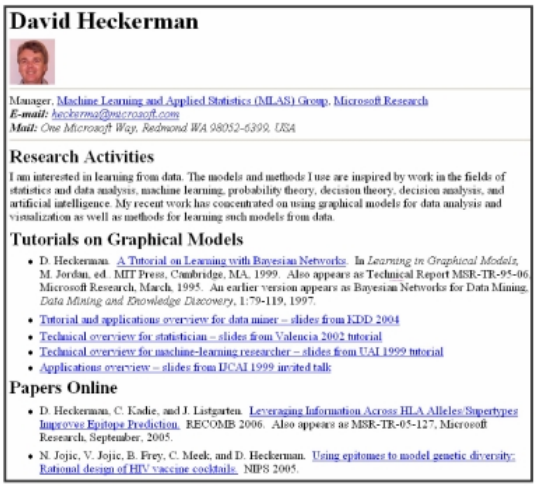

Figure 1: The homepage of David Heckerman contains his contact information (name, address, and email) and academic information (title, affiliation, papers and academic activities).

This is because the rules learned for one type of webpages could not be easily adapted to other types of webpages. Thus, template-independent methods are required in this type of webpage understanding tasks.

2. The attribute values of a researcher are presented in multiple separated HTML elements. For example, the attributes (title, author, publication conference and year) of the first paper record in Fig. 1 are presented in three HTML elements which are visually close to each other in the webpage. Thus, understanding of page structure is required to calculate the visual distance between these HTML elements and group the related information together.

3. The text content of a single HTML element could contain information of multiple attributes. For example, in Fig. 1 all the author names of a paper are presented in a single HTML element, and several address-related attributes (street, city, state, and zip code) are also presented in a single HTML element. For convenience, we use "text fragment" to refer to the text content of a single HTML element. Thus, to identify the attribute values of a researcher, understanding (including segmenting and labeling) the text fragments is required.

Thus, an effective webpage understanding model should be template-independent, and the segmentation and labeling of both page structure and text fragments are required. Existing work on web mining and natural language processing is ineffective due to the following two reasons.

Firstly, existing work on web mining mainly focus on page layout and format analysis using rule-based pattern mining approaches [1][8][26][27]. Little work has been done to effectively process the text fragments except some wrapper-based (or rule-based) approaches [14][17]. However, wrapper-based approaches could not solve our problem because they are template-dependent and could only work for webpages generated by the same template.

Secondly, existing work on natural language processing cannot be directly applied to web text understanding. Because the text contents on webpages are often not as regular as those in natural language documents and many of them are less grammatical text fragments. One possible method of using NLP techniques for web text understanding is to first manually or automatically identify logically coherent data blocks, and then concatenate the text fragments within each block into one string via some pre-defined ordering method. The concatenated strings are finally put into a text processing method, such as CRYSTAL [22] or Semi-Markov Models [20], to identify target information. [22][9] are two attempts in this direction. This type of pre-processing based approaches has several disadvantages:

Ineffectiveness in Rule-based Webpage Segmentation: The segmentation of logically coherent data blocks is non-trivial for rule-based approaches because of the diversity of webpages. It is demonstrated in [29] that de-coupled approaches to detecting data records without semantics of the contents are highly ineffective.

Insufficient Grammar for NLP Systems: Even if the logically coherent data blocks could be identified correctly, the parsed text fragments within these blocks are still lack of grammars. For example, the concatenation of an anchor text with a phone or an email is apparently of no meaning in NLP systems, which typically expect grammatical sentences.

Loss of Structure and Boundary in Concatenation: The concatenation removes or softens the boundaries of different text fragments. More importantly, it also removes structure formats of the HTML elements such as two-dimensional layout information and hierarchical organization. These structural patterns have been shown to be very useful for page structure understanding [29][28].

Now, a natural question is why there is no existing work to effectively incorporate the work from the web mining community and the natural language processing community. We believe that the answer lies in the fact that most existing page structure understanding methods are rule-based but statistical models have been the theme of text processing over the last decades [18]. For this reason, existing page structure understanding methods cannot be easily merged with statistical NLP methods in a principled manner.

Our recent work has shown that statistically regular dependency patterns among the HTML elements of a webpage, such as two-dimensional [29] and hierarchical [29] dependencies, are ubiquitous. Thus, sophisticated probabilistic models, such as Hidden Markov Models or Conditional Random Fields [15], can be developed to exploit these useful dependencies for effective webpage structure understanding. However, in these approaches, we only assign semantic labels to HTML elements, and do not segment and label the text content within an HTML element.

In this paper, we study how to explore the structural dependency patterns as shown in [29] to help understand text fragments in a principled way. Instead of using simple heuristic-based pre-processing, we propose a joint webpage understanding model to do structure understanding and text content understanding simultaneously. The joint model is an integration of Hierarchical Conditional Random Fields (HCRFs) [29] and Semi-Markov Conditional Random Fields (Semi-CRFs) [20]. It can be seen as an undirected generalization of the Switching Hidden Semi-Markov Model (S-HSMM) [10]. Two differences exist between S-HSMM and our model. First, our model is discriminative but S-HSMM

is generative. Generally, discriminative models can incorporate arbitrary features of the observations, but generative models must make some strong independence assumption to achieve inference tractability. Second, the S-HSMM used in [19][10] is a two-layer hierarchical model. Thus, it is not sufficient for webpage understanding as webpages are arbitrary hierarchical trees.

As our model performs both page structure understanding and text content understanding together, it can take raw webpages as inputs and identifies the desired information if exists. Thus, it is an integrated solution of webpage understanding. In contrast, [22][9] are de-coupled approaches.

Specifically, we make the following contributions:

1. We are the first to incorporate both structure and text content understanding into one probabilistic model for integrated webpage understanding. We show that structures can be explored to help understand text contents in a principled manner.

2. We present an undirected graphical model which is an integration of HCRFs and Semi-CRFs. The joint model can be viewed as a discriminative generalization of the Switching Hidden Semi-Markov Model [10].

3. An empirical study of our model on the task of homepage information extraction is presented.

The rest of the paper is organized as follows. In the next section, an overview of our approach is presented to help understand how the new approach works. Section 3 formally describes the proposed model. Section 4 presents our empirical studies and discussions. Section 5 discusses related work and section 6 brings this paper to a conclusion. Finally, we give our acknowledgements in section 7.

2. OVERVIEW OF OUR APPROACH

For webpage understanding, choosing a good representation of webpages is important. Tag-trees, which are natural representations of the tag structures, are commonly used in the literature. However, tag-trees tend to reveal presentation structure rather than content structure. We need a webpage representation which can effectively keep related contents together while separating semantically different blocks. As shown in [29], vision-trees are the best representation available for webpage understanding. Vision-trees are built using a vision-based page segmentation approach, which makes use of page layout features such as font, color, and size to keep HTML elements with related contents together. We use the vision tree of a webpage as its representation format. Each node on a vision-tree represents a data region (or a block) in the webpage. The root block represents the whole page. The leaf blocks are the HTML elements of the webpage. Each inner block is the aggregation of all its child blocks. The individual tokens inside the text leaf nodes (i.e. text fragments) are atomic units for text content understanding.

Based on the vision-tree, we propose a joint model by integrating HCRFs [29] and Semi-CRFs [20] to explore structural regularities to help understand text fragments. Fig. 2 shows the model structure. At coarse levels (i.e. the blocks of the vision-tree) it is a full HCRF model for page structure understanding as in [29], and at the finest level (i.e. the text contents of the leaf nodes) a Semi-CRF model is

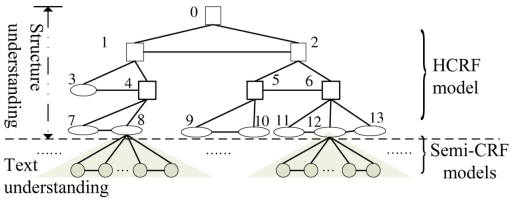

Figure 2: Graph of the joint model. The upper part is an HCRF model for structure understanding, and the models in triangles descending from leaf nodes are Semi-CRF models for text understanding.

introduced for segmenting and labeling the text fragment of each HTML element. To be integrated with the upper hierarchical model, the label assignments of leaf nodes must be incorporated into the segmentation[1] of a text fragment in the Semi-CRF models, that is, for Semi-CRF models a label assignment is a combination of both the segmentation of text fragments and the label assignments of the variables at leaf nodes. We will refer to these Semi-CRF models as extended Semi-CRF models in our joint approach. We assume that the Semi-CRF models at different leaf blocks share the same set of feature functions and weights. We show that although the model is an integration of two different types of models, the maximum likelihood estimation can be carried out separately. Thus, existing algorithms are sufficient for the training. The most likely label assignments of the variables in HCRF and the most likely segmentation and labeling of text fragments can be found jointly. However, exact joint algorithms can be too expensive for large-scale webpage understanding. Alternatively, we adopt an efficient de-coupled approximate method by first doing structure understanding and then doing text understanding with the extended Semi-CRF models. In structure understanding the effectiveness of leveraging semantic labels in both directions has been demonstrated in [29]. Here, we focus on studying the effectiveness of leveraging semantic labels of HTML elements in text understanding.

An Illustration Example: In Fig. 1, the publication information is grouped into small data regions or paper records. Take the first paper record as an example. All the author names are presented in the first HTML element, and the title of the paper is in the second element. The publication year and conference are presented in the third element. Traditional approaches [22][9] first detect the paper record, and then process the concatenated text string of all the text elements within the paper record. The shortcomings of these methods have been discussed previously. In contrast, our approach takes an integrated procedure to identify paper records, assign semantic labels to HTML elements, and further segment text contents into purified attributes. In our model, extended Semi-CRF models are aware of the semantic labels of the text elements when doing segmentation and labeling. This will help identify the boundary of each attribute. For example, if the first element "D. Heckerman, C. Kadie, and J. Listgarten." is labeled as containing only author names by the HCRF, then the extended Semi-CRF can leverage this semantic information and easily identify the boundary of each author name. The semantic labels of HTML elements can be accurately assigned with the help of

[1]see section 3.1.2 for the formal definition.

structure understanding as shown in [29]. Thus, structure understanding can help text content understanding.

Note that we choose Semi-CRFs for text understanding, although other models like Hidden Markov Models are possible. This is because, Semi-Markov models have been examined in [5][20] and demonstrated to be among the most promising methods for text segmentation and labeling, especially for their great power in incorporating segment-based features. Furthermore, discriminative models generally have great flexibility in encoding arbitrary useful features for information extraction compared to generative models.

3. A JOINT WEBPAGE UNDERSTANDING MODEL

In this section, we first introduce some basic concepts of Conditional Random Fields [15] and their extensions - HCRFs [29] and Semi-CRFs [20]. Then, we present a joint model for integrated webpage understanding. We show that the parameter estimation of the joint model can be performed independently. We present an exact algorithm and its efficient approximation to find the maximum a posterior assignment of all the variables in the joint model.

3.1 Preliminaries

In general, Conditional Random Fields (CRFs) are Markov Random Fields that are globally conditioned on observations. Let $G = (V, E)$ be an undirected model over a set of random variables Y and X. X are variables over the observations to be labeled and Y are variables over the corresponding labels. Y can have non-trivial structures, such as linear-chain [15] and 2D grid [28]. The conditional distribution of a label assignment y (an instance of Y) given the observations x (an instance of X) has the form

$$p(\mathrm{y}|\mathrm{x}) = \frac{1}{Z(\mathrm{x})} \prod_{c \in \mathbb{C}} \psi_c(\mathrm{y}|_c, \mathrm{x}), \tag{1}$$

where $\mathbb{C}$ is the set of cliques in G; $\mathrm{y}|_c$ are the components of y associated with the clique c; ψ_c is a potential function defined on $\mathrm{y}|_c$ and takes non-negative real values; $Z(\mathrm{x}) = \sum_{\mathrm{y}} \prod_{c \in \mathbb{C}} \psi_c(\mathrm{y}|_c, \mathrm{x})$ is the normalization factor or partition function. The potential functions are expressed in terms of feature functions $f_k(\mathrm{y}|_c, \mathrm{x})$ and their weights λ_k

$$\psi_c(\mathrm{y}|_c, \mathrm{x}) = exp\Big\{ \sum_k \lambda_k f_k(\mathrm{y}|_c, \mathrm{x}) \Big\}.$$

Given a set of training samples $D = \{(\mathrm{y}^i, \mathrm{x}^i)\}_{i=0}^N$, parameter estimation is to choose the feature functions' weights to maximize the likelihood of the set of training samples. For linear-chain CRFs [15], this task can be efficiently done with dynamic programming algorithms.

3.1.1 Hierarchical Conditional Random Fields

An HCRF model (HCRFs) [29] is a CRF model but with the variables indexed by the vertices of a hierarchical graph. The probability distribution has the same form as in equation (1). Compared with the traditional linear-chain model, an HCRF model has a different set of cliques, and thus has a different set of feature functions. For HCRFs, triangles are the maximum cliques, and feature functions defined on these cliques encode the dependencies among parent variables and their children. Thus, it can capture the dependencies between the variables at adjacent levels of the graph in Fig. 2. Via the inter-level dependencies and the dependencies of the variables at the same level, HCRFs provide a way to incorporate long distance dependencies for accurate structure understanding. Another advantage of this model is that by using the standard junction tree algorithm, it is very efficient to do parameter estimation and to find the maximum a posterior label assignment. In fact, the algorithm is linear in terms of the number of elements.

3.1.2 Semi-Markov Conditional Random Fields

A Semi-CRF model (Semi-CRFs) [20] is another extension of the linear-chain CRFs for sequence data segmentation and labeling. Here, we use the same notations as in [20]. In this model, x is a token sequence and $|\mathrm{x}|$ is the sequence's length (i.e. number of tokens). The vector $\mathrm{s} = \langle s_1, s_2, \ldots, s_n \rangle$ is a segmentation of x, and each entry is a segment which is a triple $s_i = \langle t_i, \mu_i, y_i \rangle$ with t_i as a start position, μ_i as an end position, and y_i as the label of this segment. Thus, a segment s_i means that the label y_i is assigned to all the observations between the start position t_i and the end position μ_i in the observation sequence x. It is reasonable to assume that segments have positive lengths and adjacent segments touch, that is, $0 \leq t_i \leq \mu_i \leq |\mathrm{x}|$ and $t_{i+1} = \mu_i + 1$. Let g^k be a feature function, and it depends on the current segment, the whole observation, and the label of previous segment, that is, $g^k(i, \mathrm{x}, \mathrm{s}) = g^k(y_{i-1}, y_i, t_i, \mu_i, \mathrm{x})$. Let $\mathrm{g} = \langle g^1, g^2, \ldots, g^K \rangle$ be a vector of feature functions and $\mathrm{W} = \langle w_1, w_2, \ldots, w_K \rangle$ be the corresponding weight vector. As a Conditional Random Field model, the probability distribution $p(\mathrm{s}|\mathrm{x})$ is also of the form as in equation (1) but with the traditional label assignment y replaced by a segmentation s and the cliques are replaced by segments:

$$p(\mathrm{s}|\mathrm{x}) = \frac{1}{Z(\mathrm{x})} \prod_{i=1:|\mathrm{s}|} \psi_i(i, \mathrm{x}, \mathrm{s}), \tag{2}$$

where $\psi_i(i, \mathrm{x}, \mathrm{s}) = e^{\mathrm{W} \cdot \mathrm{g}(i, \mathrm{x}, \mathrm{s})}$, and $Z(\mathrm{x}) = \sum_{\mathrm{s}} \prod_{i=1}^{|\mathrm{s}|} \psi_i(i, \mathrm{x}, \mathrm{s})$.

For Semi-CRFs, parameter estimation and finding the maximum a posterior segmentation can be efficiently carried out via a dynamic programming algorithm. The computational complexity is a constant factor more than that of the traditional linear-chain model when the maximum length of the segments is assumed to be fixed. Recent work in [21] shows that the computational complexity could be further reduced by defining succinct potentials.

3.2 A Joint Model

Now, we present the integrated webpage understanding model. The model's graph is shown in Fig. 2. The upper part is a full hierarchical model with $M + 1$ nodes, and at each leaf node on the vision-tree a Semi-CRF model is introduced for text segmentation and labeling. For the hierarchical model, we use rectangles to denote inner nodes and use ellipses to denote leaf nodes. Each node on the graph is associated with a random variable Y_i, and all the variables $Y = \{Y_i\}_{i=0}^M$ are organized in a hierarchy. A label assignment of these variables $\mathrm{y} = \{y_i\}_{i=0}^M$ is organized in a hierarchy as Y. We partition the set into two subsets: $Y = \{Y_{l,i}\}_{i=0}^L \cup \{Y_{p,j}\}_{j=0}^{M-L-1}$, where $Y_{l,i}$ are variables at leaf nodes; $Y_{p,j}$ are variables at inner nodes; and $L + 1$ is the number of leaf nodes.

To be integrated with the upper hierarchical model, extensions must be made to Semi-CRF models. In the next two sections, we first describe the extensions of Semi-CRF

models to incorporate structure understanding when doing text segmentation and labeling, and then present a joint model to integrate the two parts together.

3.2.1 Extended Semi-CRF Model

In order to be integrated with the upper hierarchical model, the label assignments of the variables at leaf blocks (or leaf variables for short) must be incorporated into the segmentation of text fragments in the lower Semi-CRF models, that is, for Semi-CRF models a label assignment is a combination of both the segmentation of text fragments and the label assignments of the leaf variables. In this way, the upper hierarchical model and the lower Semi-CRF models are integrated via the leaf variables. This is the key difference from the standard Semi-CRF model [20]. We will refer to these Semi-CRF models as extended Semi-CRFs in our joint approach. Here, we assume that the extended Semi-CRFs at different leaf blocks are conditionally independent given the leaf variables at which they are located. We also assume that the Sem-CRF models at different leaf blocks share the same set of feature functions and parameters.

Now, we take one leaf block as an example to formally define the extended model. For the leaf variable $Y_{l,i}$, the extended Semi-CRF model is defined as follows. The observation sequence $\mathrm{x}_{l,i}$ is the text fragment at the leaf block. Let $\mathrm{s}_i = \langle s_{i,1}, s_{i,2}, \ldots, s_{i,n_i}\rangle$ denote a segmentation of $\mathrm{x}_{l,i}$. Here, each segment is an extension of the segment of the standard Semi-CRF model to incorporate the label assignment of the leaf variable: $s_{i,j} = \langle t_{i,j}, \mu_{i,j}, y_{i,j}, y_{l,i}\rangle$, where $t_{i,j}$ is a start position; $\mu_{i,j}$ is an end position; $y_{i,j}$ is the label of this segment; and $y_{l,i}$ is the label of the leaf variable $Y_{l,i}$. We will call these leaf labels $y_{l,i}$ as *supper labels* as opposed to the labels $y_{i,j}$ used within the extended Semi-CRFs. Correspondingly, the feature functions for the extended Semi-CRFs are also dependent on the labels of leaf variables. Let g^k be a feature function. It maps a triple $(j, \mathrm{x}, \mathrm{s}_i)$ to a real value. Here, we assume that it depends on the current segment, the whole observation, the label of previous segment, and also the label of the leaf variable $Y_{l,i}$. So, $g^k(j, \mathrm{x}, \mathrm{s}_i) = g^k(y_{i,j-1}, y_{i,j}, y_{l,i}, t_{i,j}, \mu_{i,j}, \mathrm{x})$. Let g be the vector of feature functions and W be the corresponding weight vector as defined before. Then, the conditional probability of segmentation $p(\mathrm{s}_i|\mathrm{x})$ has the same form as in (2) but with the original segmentation replaced by an extended one and the feature functions are replaced correspondingly.

3.2.2 The Joint Distribution

Now, for the joint model we define $\mathrm{S} = \langle \mathrm{s}_0, \mathrm{s}_1, \ldots, \mathrm{s}_L\rangle$ to be the segmentations of all the leaf blocks. Then, an assignment of all the variables in the joint model is a pair $\langle \mathrm{y}, \mathrm{S}\rangle$ where y is the label assignment of the upper hierarchical model and S is the segmentation assignment of the extended Semi-CRFs. A valid assignment $\langle \mathrm{y}, \mathrm{S}\rangle$ must satisfies the condition that the two assignments match at the leaf variables, that is, the label assignments of the leaf variables from both the upper hierarchical model and the lower extended Semi-CRF models are the same: $\mathrm{s}_i \cdot y_{l,i} = \mathrm{y} \cdot y_{l,i}, 0 \le i \le L$. In the following, we will use $\langle \mathrm{y}, \mathrm{S}\rangle$ to denote a valid assignment without further explanation.

Then, the joint probability distribution of our model has the following factorization form

$$p(\langle \mathrm{y}, \mathrm{S}\rangle|\mathrm{x}) = p(\mathrm{y}|\mathrm{x})p(\mathrm{S}|\mathrm{x}, \mathrm{y}) = p(\mathrm{y}|\mathrm{x}) \prod_{i=0:L} p(\mathrm{s}_i|\mathrm{x}, y_{l,i}), \quad (3)$$

where the first equation is for the chain rule, and the last equation is for the conditional independency assumption that given the label of a leaf variable, the segmentation assignment of that leaf variable is independent from the label assignment of other variables. We shall see that this factorized distribution will lead to an efficient separate parameter estimation algorithm. The joint model can be viewed as an undirected generalization of the Switching Hidden Semi-Markov Model [10] since it can be viewed as the concatenation of many Semi-CRF models and the leaf variables act as switch variables.

In equation (3), each part can be computed efficiently. For the hierarchical model, the conditional probability is expressed by the feature functions as in [29]

$$p(\mathrm{y}|\mathrm{x}) = \frac{1}{Z_h(\mathrm{x})} exp\Big\{ \sum_{v,k} \mu_k g_k(\mathrm{y}|_v, \mathrm{x}) + \sum_{e,k} \lambda_k f_k(\mathrm{y}|_e, \mathrm{x}) + \sum_{t,k} \gamma_k h_k(\mathrm{y}|_t, \mathrm{x})\Big\},$$

where g_k, f_k, and h_k are feature functions defined on three types of cliques (i.e. vertex, edge, and triangle) respectively; μ_k, λ_k, and γ_k are the corresponding weights; $v \in V$, $e \in E$, and t is a triangle. $Z_h(\mathrm{x})$ is the normalization factor of the hierarchical model.

For the extended Semi-CRF model, the conditional probability is

$$p(\mathrm{s}_i|\mathrm{x}, y_{l,i}) = \frac{1}{Z_i(\mathrm{x}, y_{l,i})} exp\Big\{ \sum_{j,k} \omega_k g^k(j, \mathrm{x}, \mathrm{s}_i)\Big\},$$

and the normalization factor is

$$Z_i(\mathrm{x}, y_{l,i}) = \sum_{\mathrm{s}'_i : \mathrm{s}'_i \cdot y_{l,i} = y_{l,i}} exp\Big\{ \sum_{j,k} \omega_k g^k(j, \mathrm{x}, \mathrm{s}'_i)\Big\}.$$

3.3 Parameter Estimation

For the joint model, each training sample in D is a pair $(\langle \mathrm{y}, \mathrm{S}\rangle^i, \mathrm{x}^i)$ and the log-likelihood function is

$$\mathcal{L}(\Theta, \mathrm{W}) = \sum_{i=0:N} \log p(\langle \mathrm{y}, \mathrm{S}\rangle^i|\mathrm{x}^i, \Theta, \mathrm{W}),$$

where $\Theta = \langle \mu_1, \mu_2, \ldots; \lambda_1, \lambda_2, \ldots; \gamma_1, \gamma_2, \ldots\rangle$ is the parameter vector of the hierarchical model and W is the parameter vector of the extended Semi-CRF models. Substitute the distribution in (3) into the log-likelihood and we get

$$\begin{aligned}\mathcal{L}(\Theta, \mathrm{W}) &= \sum_{i=0:N} \log p(\mathrm{y}^i|\mathrm{x}^i, \Theta, \mathrm{W}) + \sum_{i=0:N} \log p(\mathrm{S}^i|\mathrm{x}^i, \mathrm{y}^i, \Theta, \mathrm{W}) \\ &= \mathcal{L}_h(\Theta, \mathrm{W}) + \sum_{i=0:N} \sum_{j=0:L^i} \log p(\mathrm{s}^i_j|\mathrm{x}^i, y^i_{l,j}, \Theta, \mathrm{W}) \\ &= \mathcal{L}_h(\Theta) + \mathcal{L}_s(\mathrm{W}).\end{aligned}$$

The last equation is due to the fact that parameters Θ and W are independent. Thus, the maximization of $\mathcal{L}(\Theta, \mathrm{W})$ is equivalent to the maximization of $\mathcal{L}_h(\Theta)$ and the maximization of $\mathcal{L}_s(\mathrm{W})$. Now, we can perform the parameter estimation for the HCRF model and the extended Semi-CRF models separately. Here, we use the algorithm in [29] to train the hierarchical model. For the extended Semi-CRF model, similar dynamic programming algorithms as in [20] can be used to learn the parameters, but with the segmentation replaced by the extended one in order to incorporate *supper labels*. To compute the normalization factor $Z_i(\mathrm{x}, y_{l,i})$,

forward vectors can be recursively defined as

$$\alpha(j, y, y_{l,i}) = \sum_{d=1:M} \sum_{y'} \alpha(j-d, y', y_{l,i}) e^{\mathrm{W}\cdot \mathrm{g}(y', y, y_{l,i}, j-d, j, \mathrm{x})}$$

with the base case $\alpha(0, y, y_{l,i}) = 1$. Here, M is the maximum segment length [20]. The normalization factor is $Z_i(\mathrm{x}, y_{l,i}) = \sum_y \alpha(|\mathrm{x}_{l,i}|, y, y_{l,i})$. Similarly, we define the recursion as

$$\eta^k(j, y, y_{l,i}) = \sum_{d=1:M} \sum_{y'} \beta^k(y', y, y_{l,i}, j-d, j, \mathrm{x}) e^{\mathrm{W}\cdot \mathrm{g}(y', y, y_{l,i}, j-d, j, \mathrm{x})},$$

where

$$\beta^k(y', y, y_{l,i}, j-d, j, \mathrm{x}) = \eta^k(j-d, y', y_{l,i}) + \alpha(j-d, y', y_{l,i}) g^k(y', y, y_{l,i}, j-d, j, \mathrm{x}).$$

Then, the expectation of feature function g^k with respect to the model distribution can be computed using the same formula as in [20] but with the normalization factor and $\eta^k(j, y)$ replaced by $Z_i(\mathrm{x}, y_{l,i})$ and $\eta^k(j, y, y_{l,i})$ respectively. To avoid over-fitting, the spherical Gaussian prior with mean $\mu = 0$ and variance matrix $\Sigma = \delta^2 I$ is used to penalize the log-likelihood when training each part of the model.

3.4 Finding the Most Likely Assignment

For webpage understanding, the target is to find the best assignment of the variables in the model, that is, the pair $\langle \mathrm{y}, \mathrm{S} \rangle$ that has the maximum posterior probability. We have shown that parameter estimation can be performed independently for two different parts without loss of accuracy. But finding the maximum a posterior assignment is not the case. This is because unlike training webpages, there are no 'true' labels assigned to the leaf nodes of a testing webpage. Thus, all the possible assignments to a leaf variable must be computed.

Based on the junction tree algorithm [29], we can develop a joint optimization algorithm to find the most likely assignment. We can take the same procedure as in [29] to construct a junction tree for the upper hierarchical model, and then propagate messages on the constructed junction tree using the two-phase schedule algorithm [13]. Before running the schedule algorithm, the messages from the extended Semi-CRF models must be integrated in order to incorporate the effects of text content understanding into structure understanding. Here, the local messages at the leaf variable $Y_{l,i}$ are the normalization factor $Z_i(\mathrm{x}, y_{l,i})$ which can be recursively computed using the forward algorithm as above. The local messages are incorporated by multiplying them into the initial potentials on the constructed junction tree. After initialization, the two-phase schedule algorithm runs to find the most likely label assignments of all the variables at inner nodes. At the end of the schedule algorithm, the marginal potentials of the leaf variables are incorporated into the extended Semi-CRF models to find the best label assignments of the leaf variables and also the best segmentation of text fragments. This can be done using Viterbi algorithm by defining the recursion

$$V(j, y, y_{l,i}) = \max_{d=1:M, y'} (V(j-d, y', y_{l,i}) + \mathrm{W}\cdot \mathrm{g}(y', y, y_{l,i}, j-d, j, \mathrm{x}))$$

with the base $V(0, y, y_{l,i}) = 0$. Then, the most likely label assignment of the leaf variable is the label $y^\star_{l,i}$ that achieves the highest value $\max_{y, y_{l,i}} (V(|\mathrm{x}_{l,i}|, y, y_{l,i}) + \phi(y_{l,i}))$ and the best segmentation is the path traced by $\max_y V(|\mathrm{x}_{l,i}|, y, y^\star_{l,i})$, where $\phi(y_{l,i})$ is the marginal potential of the variable at the end of the two-phase algorithm.

However, the joint optimization algorithm can be too expensive for large-scale webpage understanding because for each label assignment of a leaf variable the dynamic programming algorithm must be run twice to collect messages and to find the best segmentation of its content. An alternative efficient approximate method is to find the maximum a posterior assignment of two parts separately. First, the most likely assignment of the variables in the upper hierarchical model is found using the junction tree algorithm [29]. Then, leaf blocks together with supper labels are further segmented using the extended Semi-CRF models. During the segmentation and labeling of text fragments, the supper labels of leaf blocks are fixed. Let $y^\star_{l,i}$ be the most likely label assignment of the leaf variable $Y_{l,i}$ at the end of the first step, we compute the following recursion

$$V(j, y, y^\star_{l,i}) = \max_{d=1:M, y'} (V(j-d, y', y^\star_{l,i}) + \mathrm{W}\cdot \mathrm{g}(y', y, y^\star_{l,i}, j-d, j, \mathrm{x}))$$

with the base $V(0, y, y^\star_{l,i}) = 0$. Then, the best segmentation is the path traced by $\max_y V(|\mathrm{x}_{l,i}|, y, y^\star_{l,i})$.

In this approximate algorithm, although the segmentation of text contents is not considered when doing structure understanding, the semantic labels assigned by the upper hierarchical model are considered to help text understanding. One advantage of this approximate algorithm is that after the first step of structure understanding, most of the text fragments are labeled as containing no interested information. These non-informative fragments can be kept away from further segmentation and labeling. This simple heuristic could significantly reduce the time complexity because most contents on webpages contain no interested information but act as decorations, supplements or something else.

4. EXPERIMENTS

In this section, we report empirical results by applying our proposed model to understand web text fragments and identify structured information from researchers' homepages. We compare our proposed model with a sequential approach that combines the state-of-the-art algorithms in record detection and text segmentation. The results show that our model achieves significant improvements in the final understanding of text fragments through leveraging the semantic labels of page structures. We also study the effects of NLP features and database features if they are available.

4.1 Dataset

Although some datasets like the *Job* corpus [3] and the *Address and Paper* corpora [2] have been evaluated in previous work, they are raw text documents and do not have HTML structures. Thus, they are not suitable for our evaluation since our focus is on both structures and text contents. Also our work is different from [2][5], in which the inputs are some pre-identified segments of a webpage such as address records and paper records which are treated as string sequences in their experiments. In our method, we take raw pages as inputs and automate both the identification of address and paper records and the segmentation and labeling of the text fragments.

As we have stated, extracting structured information from researchers' homepages needs both structure understanding

and text content segmentation and labeling. So, we evaluate our models on this task. We setup our dataset with 292 homepages of computer science researchers. We identify both academic and contact information of a researcher. For contact information, we identify his/her *Name*, address (including *Street*, *City*, *State*, and *Zip code*), *Phone*, and *Email*. For academic information, we extract his/her academic *Title* (like professor, lecturer, and etc.), *Affiliation*, and publications. For each paper, we identify *Paper Title*, *Author* name, *Publisher* type (like conference, workshop, journal, PhD thesis, and etc.), and publication *Year*.

The 292 homepages are randomly downloaded from the computer science departments of about 10 American Universities including Stanford, MIT, CMU, and UC Berkeley, and also some homepages are from research labs like Bell labs, Microsoft Research, IBM Research, and etc. All the pages are manually labeled and all the structured information is segmented. Statistics of the dataset is shown in the first row of Table 1. You may note that the number of owner names is less than the number of homepages. This is because there are four names presented in images, and one page without any explicit name.

4.2 Methods and Evaluation Metrics

To our knowledge, there is no complete solution for webpage understanding (both structure understanding and text understanding). The heuristic-based pre-processing methods [9][22] are not appropriate baseline methods because their rules are used for specific types of problems, and it is difficult to develop an optimal set of rules in our experiments. Here, we build our baseline method by first using the most recently developed webpage segmentation and labeling method [29] to identify interested text fragments and then applying Semi-CRF models [20] to segment text fragments and identify structured information. When being input into Semi-CRF models, text fragments are concatenated into strings as in [20]. Instead of concatenating all the text fragments on a webpage into one single string, which is apparently inefficient, we concatenate the fragments within each paper record as one string. The text fragments related to *Street*, *City*, *Region*, *Zip code*, *Email*, and *Phone* are concatenated together, and the other text fragments including the owner's *Name*, *Title*, *Affiliation* and non-informative text fragments are concatenated together. We refer to this method as sequential baseline method in the sequel.

As we defined in [29], there are two types of label spaces for HCRFs: leaf label space and inner label space. The leaf label space is used for labeling the leaf nodes of the vision-tree. It consists of all the target attributes, and the combinations: *Name and Title*; *Title and Affiliation*; *Name, Title and Affiliation*; *City and Region*; *Region and Zip code*; *City, Region and Zip code*; *Paper Title and Author*; *Author and Publisher*; *Paper Title and Publisher*; *Publisher and Year*; *Paper Title, Author and Publisher*. Note that here we include the combinations in leaf label space. This is because the text fragments which are labeled by HCRFs can contain multiple attributes. The inner label space is used for labeling the inner blocks of the vision-tree. It consists of *Paper Record*, *Paper List*, *Note block*, and all the above combinations. For the extended Semi-CRF models, there are also two types of label spaces: supper label space and attribute label space. The attribute label space consists of the names of the attributes we are interested in. The supper label space is the same as the leaf label space of the HCRF model. When finding the maximum a posterior segmentation and labeling using the extended Semi-CRF models, the upper bounds on segment lengths are dependent on the supper labels. For *Title*, *Affiliation*, *Street*, *Paper Title*, and *Publisher*, the upper bounds (ranging from 10 to 14) are larger than those for *Name*, *City*, *State*, *Zip code*, *Email*, *Phone*, *Author*, and *Year* (ranging from 2 to 4).

For each attribute, the standard *P*recision, *R*ecall and *F1* measure are evaluated.

4.3 Results and Discussions

In our experiment, the dataset is partitioned into two subsets: 50 percent for training and 50 percent for evaluation. For the training of the HCRF model and the extended Semi-CRF model, spherical Gaussian priors are used with the same standard variance 5.

The performance is shown in Table 1. We can see that the integrated model significantly outperforms the sequential baseline method on all the attributes except *Zip code*, *City*, *Region*, *Phone*, and *Year*. The performance on *Zip code* is almost the same, and the baseline method can perform well on *City*, *Region*, and *Phone*. This is because for these attributes, text patterns are usually very distinctive, so the baseline method can perform well even without consideration of where they appear in a webpage. Also relatively clean presentation of contact information in homepages makes it easy to distinguish useful information from noise. For *Year*, after the first step of identifying paper records, they can be easily and accurately identified with the baseline method.

However, for other attributes no simple features can be used to effectively identify them. For example, both *Title* and *Affiliation* can have similar patterns (such as the first letters of words are all capitalized, in bold font, and etc.) as those of *Name*. For paper related attributes, *Publisher* can appear like a *Paper Title*, and *Author* can appear in front of or after *Paper Title*. This leads to more ambiguities when identifying each of them. In these cases, structural layout patterns are more helpful. In our proposed model, the upper hierarchical model can effectively incorporate hierarchical dependency patterns and also long distance dependencies to understand the semantics of text fragments. Then, the semantic labels assigned by upper HCRF model are incorporated into the lower extended Semi-CRF models when doing segmentation and labeling of text fragments. This procedure can help the final identification of our interested attributes. For example, if the HCRF tells that a text fragment is a list of author names even though it doesn't tell where the boundaries between different author names are, then it will be much easier for the lower Semi-CRF model to identify the unknown boundaries.

From the results, we can also see that in researchers' homepages, *Name*s are much easier to be found compared with *Title* and *Affiliation*. So the sequential baseline method can perform well in identifying *Name*. The lower performance on *Email* compared to that on *City*, *Region*, and *Zip code* is due to the fact that researchers often make their email addresses unreadable to automated programs. For *Title*, *Affiliation*, *Region*, and *Zip code*, the baseline method achieves higher recall and lower precision compared to the integrated model. This is because there are usually some false-positives on webpages which are wrongly detected by the baseline method because of its lack of structural infor-

Table 1: Performance of our model with and without NP-chunking pre-processing, and the sequential baseline model. Here, we use Aff, Reg, Zip, Pub to denote Affiliation, Region, Zip Code, and Publisher respectively.

Attributes		Name	Title	Aff	Street	City	Reg	Zip	Phone	Email	Paper Title	Author	Pub	Year
Total Number		287	295	417	315	189	189	194	217	207	1457	4032	1382	1282
Integrated Model	*P*	0.939	0.762	0.704	**0.812**	**0.946**	**0.935**	**0.978**	**0.860**	**0.805**	**0.918**	0.843	0.725	0.906
	R	0.870	0.671	0.847	0.574	**0.931**	0.915	0.928	**0.903**	**0.855**	**0.844**	**0.844**	**0.698**	**0.854**
	F1	0.903	0.714	0.769	**0.673**	**0.938**	**0.925**	0.952	**0.881**	**0.829**	**0.879**	**0.843**	**0.711**	**0.879**
Sequential Baseline	*P*	0.879	0.429	0.386	0.344	0.774	0.730	0.954	0.805	0.668	0.611	0.829	**0.786**	**0.933**
	R	0.817	**0.722**	**0.993**	**0.602**	0.926	**0.931**	**0.969**	0.857	0.778	0.479	0.509	0.457	0.746
	F1	0.847	0.538	0.556	0.438	0.843	0.818	**0.961**	0.830	0.719	0.537	0.631	0.577	0.829
Non-NP Chunking	*P*	**0.947**	**0.814**	**0.716**	0.809	0.936	**0.935**	**0.978**	0.856	0.794	0.912	**0.867**	0.721	0.907
	R	**0.873**	0.637	0.842	0.574	0.926	0.915	0.928	**0.903**	**0.855**	0.833	0.677	0.685	0.850
	F1	**0.908**	**0.715**	**0.774**	0.672	0.931	**0.925**	0.952	0.879	0.823	0.871	0.760	0.703	0.878

mation. But the integrated model can identify this noise information. Of course, some true-positives are missed by the integrated model. As we shall see in the next section, another part of the improvements is from the incorporation of noun phrase chunking.

Other reasons for the promising performance of our model include the incorporation of long distance dependencies in HCRFs [29] and also global features that are extracted from the alignment of the paper records in the same page. The alignment is based on the observation that each researcher always presents his publications in one similar pattern, although the patterns may be different for different researchers.

4.3.1 Incorporation of NP-Chunking

Much work has been done to investigate the usability of shallow or deep linguistic structures for various application tasks such as named entity extraction and language chunking [7]. In contrast to deep natural language processing, shallow NLP techniques are more robust and more efficient. This is very important for scalable webpage understanding. Thus, in our experiment we incorporate the noun phrase chunking (NP-Chunking) results of a fast chunk parsing method [25]. All the text fragments on the homepages in our dataset are parsed using the method [25]. To control the bad effects of incorrect chunking, we only use the noun phrases that are at the finest level of the parsing trees [25]. We treat the tokens within one noun phrase as an individual unit in Semi-CRF models during segmentation. Thus, they appear together in one unit in the final results. Tokens that are not in noun phrases are treated the same as in the approach without NP-Chunking.

The performance of our method without NP-Chunking is shown in the last row of Table 1. We can see that our method can also perform better than the baseline method even without NP-Chunking. For our model with or without NP-Chunking, the performance of most of the attributes does not change much. However, the F1 of the attribute *Author* decreases by more than 8 points when NP-Chunking is not used. This is because the method [25] can always accurately identify the boundaries between different author names when they appear in the same elements, and each author name is identified as a noun phrase. Thus, the additional constraints brought by noun phrases can help Semi-CRFs separate different author names without introducing notable errors.

4.3.2 Incorporation of External Dictionary Features

One advantage of Semi-CRFs is that they can effectively incorporate segment-based features [20]. We use the existing database DBLP (http://dblp.uni-trier.de/xml/) to define these additional features. DBLP is a public and relatively clean dataset. The total number of paper records is 0.72 million, and the number of author names is 0.48 million. Here, we evaluate the performance on *Paper Title* and *Author*. To extract database features, we adopt a strict strategy of string matching. For *Paper Title* matching, the number of matched tokens is no less than 3, and the matched tokens are kept in sequence, and also there are no punctuations appearing between the matched tokens. For *Author* matching, the number of matched tokens is no less than 2 and no more than 4, and the first letters of all the matched tokens are capitalized. By matching of two tokens we mean the exact matching of their text characters. In our dataset, about 40 percent of paper titles and about 80 percent of authors are matched with those in the DBLP database.

To see the effect of database features, we setup different subsets by randomly sampling the whole DBLP dataset with different number of matched paper titles and authors. Different settings are in Table 2. For the setting #0, zero percent of both the matched paper titles and authors means that we do not use any database features. For #16, all the matched paper titles and authors are used. Fig. 3 shows the performance of the three different methods on *Paper Title* and *Author* with different settings as in Table 2. For other attributes (*Publisher* and *Year*) the performance does not change much. From the results, we can see that when incorporating database features, the overall performance can be improved. For our method with NP-Chunking, about 3 points are achieved in F1 measure for the attribute *Author*, and almost no difference for *Paper Title*. However, for the baseline method, the performance on both *Paper Title* and *Author* is significantly improved. Also the improve-

Table 2: The matching ratios of Paper Title and Author in different sampled DBLP datasets.

#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Paper Title (%)	0	10	10	10	10	20	20	20	20	30	30	30	30	40	40	40	40
Author (%)	0	20	40	60	80	20	40	60	80	20	40	60	80	20	40	60	80

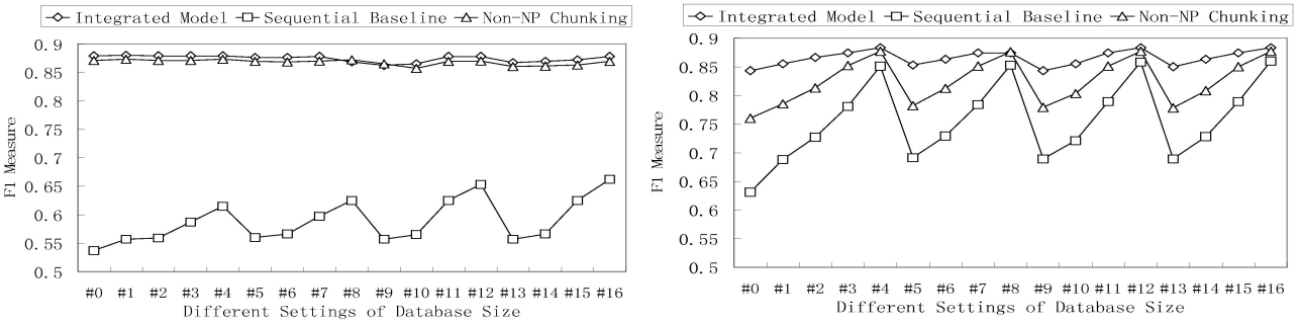

Figure 3: Performance of three methods with different ratios of matching with DBLP dataset. The left chart is the F1 measure of the attribute Paper Title, and the right one is the F1 measure of Author.

ments increase when the matching ratios increase. For our method without NP-Chunking, the performance on *Author* can be improved much. In summary, the results show that our method can also achieve very promising results when no database or only a small one is available, but the baseline method is quite dependent on the availability of databases.

4.3.3 Time Complexity

For webpage understanding, efficiency is an important issue. As we have stated, HCRFs admit an efficient inference algorithm which is linear in terms of the number of HTML elements. The inference of Semi-CRFs, which is quadratic in terms of the maximum segment length, is more expensive. But in practice the maximum lengths are usually not very large. Furthermore, the maximum lengths in the de-coupled algorithm are different for different supper labels. As in section 4.2 for many attributes, the maximum lengths are quite small. Thus, the inference algorithm is efficient. The average running time over all testing webpages is less than 2.6s, and the average time over all paper records is less than 0.3s. It means that about 33 thousands of webpages or 288 thousands of paper records can be processed by one machine in one day. Recent work in [21] shows that the computational efficiency of Semi-CRFs could be further improved. We plan to implement this method in the future.

5. RELATED WORK

Typical natural language processing methods expect fully grammatical sentences. However, the attributes on webpages are often presented in text fragments which always have some structures but are less in grammar. This makes the application of NLP methods much challenging to mine web data. Several attempts [22][12][9] have been made to apply NLP methods on the web by incorporating structure information. [22] proposes the Webfoot to segment webpages into logically coherent segments and then applied NLP methods to extract information from non-grammatical webpages. Similarly, [9] first builds an HTML Struct Tree based on page layout information, and then encodes extraction rules based on this tree representation. However, the heuristic-based methods have several disadvantages as discussed in the introduction. Another problem is that [9] only identifies the text fragments that contain the target information. Thus, it is not sufficient for webpage understanding. [12] uses query strings to identify the locations of candidate extractions and then composes extraction rules with both content and structure information for different types of formats, such as enumerations, lists, and tables. This method can't be used for other data that don't have these formats.

[29][28] are statistical webpage structure understanding methods. In [29][28] we show that inner-page layout does have statistically regular patterns such as two-dimensional dependencies within small data records and hierarchical dependencies in the whole webpage. We also show that these structural regularities can be explicitly explored in a statistical model for structure understanding tasks such as record detection and labeling of HTLM elements. However, these methods are not sufficient for webpage understanding due to their lack of capacity for text content understanding.

[3][11][23] are wrapper induction methods which depend on the types of webpages. WHISK [23] extracts information from structured, semi-structured and free text, we focus on incorporating both structures and text contents for webpage understanding. Probabilistic graphical models can take the advantage of mutual dependencies of different attributes for multiple slot information extraction. SRV [11] and RAPIER [3] extract only isolated slots. Thus, they lose the mutual dependencies of multiple attributes, and post-processing must be performed to re-assemble related information into records.

A probabilistic method is proposed in [2] by extending the HMM model. However, the inputs in their method are address or bibliography records which are already collected in a warehouse. But the inputs to our method are raw webpages. Furthermore, the webpages here have structures, but the records in [2] are represented as string sequences. [5][20]

are also for sequence data segmentation and labeling.

Attempts to incorporate hierarchical Markov models and Semi-Markov models have been made in [19][10]. Our model can be viewed as a discriminative generalization of the Switching Hidden Semi-Markov Model, and the differences between these models and ours have been discussed.

6. CONCLUSIONS

In this paper, we present an integrated model to incorporate both structure understanding and text content understanding for effective webpage understanding. To the best of our knowledge, our model is the first integrated webpage understanding model. The joint model is an integration of Hierarchical Conditional Random Fields and Semi-Markov Conditional Random Fields. At higher levels, a full hierarchical model is used to effectively incorporate structural dependency patterns for page structure understanding; and at the finest level Semi-Markov models are introduced to explore the dependencies of target attributes for effective text content understanding. Although the model is an integration of two different types of models, we show that it can be efficiently learned by separately learning each model. The feasibility and promise of our approach is demonstrated on a real-world webpage understanding task.

7. ACKNOWLEDGMENTS

The authors Jun Zhu and Bo Zhang are supported by the National Natural Science Foundation of China, Grant No. 60621062, and the National Key Foundation R&D Project, Grant No. 2003CB317007 and 2004CB318108.

8. REFERENCES

[1] A. Arasu and H. Garcia-Molina. Extracting Structured Data from Webpages. *Proc. of SIGMOD*, 2003.

[2] V. Borkar, K. Deshmukh and S. Sarawagi. Automatic segmentation of text into structured records. *Proc. of SIGMOD*, 2001.

[3] M. E. Califf and R. J. Mooney. Bottom-up relational learning of pattern matching rules for information extraction. Journal of Machine Learning Research, 2004.

[4] C.-H. Chang and S.-L. Liu. IEPAD: Information Extraction Based on Pattern Discovery. *Proc. of WWW*, 2001.

[5] W. W. Cohen and S. Sarawagi. Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Markov Extraction Processes and Data Integration Methods. *Proc. of SIGKDD*, 2004.

[6] V. Crescenzi, G. Mecca and P. Merialdo. ROADRUNNER: Towards Automatic Data Extraction from Large Web Sites. *Proc. of VLDB*, 2001.

[7] B. Crysmann, A. Frank, B. Kiefer, S. Muller, G. Neumann, J. Piskorski, U. Schafer, M. Siegel, H. Uszkoreit, F. Xu, M. Becher and H-U. Krieger. An Integrated Architecture for Shallow and Deep Processing. *Proc. of ACL*, 2004.

[8] D. W. Embley, Y. Jiang and Y.-K. Ng. Record-Boundary Discovery in Web Documents. *Proc. of SIGMOD*, 1999.

[9] D. DiPasquo. Using HTML Formatting to Aid in Natural Language Processing on the World Wide Web. Senior Honors Thesis, Carnegie Mellon University, 1998.

[10] T. Duong, H. Bui, D. Phung and S. Venkatesh. Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model. *Proc. of CVPR*, 2005.

[11] D. Freitag. Information Extraction from HTML: Application of a General Machine Learning Approach. *Proc. of AAAI*, 1998.

[12] C. Jacquemin and C. Bush. Combining Lexical and Formatting Cues for Named Entity Acquisition from the Web. *Proc. of the Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, 2000.

[13] F. V. Jensen, S. L. Lauritzen and K. G. Olesen. Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly*, 4:269-82, 1990.

[14] N. Kushmerick. Wrapper induction: efficiency and expressiveness. Artificial Intelligence, 118:15-68, 2000.

[15] J. Lafferty, A. McCallum and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. of ICML*, 2001.

[16] K. Lerman, L. Getoor, S. Minton and C. Knoblock. Using the Structure of Web Sites for Automatic Segmentation of Tables. *Proc. of SIGMOD*, 2004.

[17] K. Lerman, S. Minton and C. Knoblock. Wrapper maintenance: A machine learning approach. *Journal of Artificial Intelligence Research*, 18:149-181, 2003.

[18] C. Manning and H. Schutze. Foundations of Statistical Natural Language Processing. The MIT Press Cambridge, MA, May, 1999.

[19] D. Phung, T. Duong, S. Venkatesh and H. Bui. Topic Transition Detection Using Hierarchical Hidden Markov and Semi-Markov Models. *Proc. of MM*, 2005.

[20] S. Sarawagi and W. W. Cohen. Semi-Markov Conditional Random Fields for Information Extraction. *Proc. of NIPS*, 2004.

[21] S. Sarawagi. Efficient Inference on Sequence Segmentation Models. *Proc. of ICML*, 2006.

[22] S. Soderland. Learning to Extract Text-based Information from the World Wide Web. *Proc. of SIGKDD*, 1997.

[23] S. Soderland. Learning Information Extraction Rules for Semi-structured and Free Text. *Journal of Machine Learning*, 1999.

[24] F. Suchanek, G. Ifrim and G. Weikum. Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents. *Proc. of SIGKDD*, 2006.

[25] T. Yoshimasa and T. Jun'ichi. Chunk Parsing Revisited. *Proc. of the 9th International Workshop on Parsing Technologies*, 2005.

[26] Y. Zhai and B. Liu. Web Data Extraction Based on Partial Tree Alignment. *Proc. of WWW*, 2005.

[27] H. Zhao, W. Meng, Z. Wu, V. Raghavan and C. Yu. Fully Automatic Wrapper Generation for Search Engines. *Proc. of WWW*, 2005.

[28] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang and W.-Y. Ma. 2D Conditional Random Fields for Web Information Extraction. *Proc. of ICML*, 2005.

[29] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang and W.-Y. Ma. Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. *Proc. of SIGKDD*, 2006.