Academic Conference Homepage Understanding Using Constrained Hierarchical Conditional Random Fields

Xin Xin, Juanzi Li, Jie Tang
Department of Computer Science & Technology
Tsinghua University
Beijing, China
{xinxin, ljz, tangjie}@keg.cs.tsinghua.edu.cn

Qiong Luo
Department of Computer Science & Engineering
Hong Kong University of Science and Technology
Hong Kong, China
luo@cse.ust.hk

ABSTRACT

We address the problem of academic conference homepage understanding for the Semantic Web. This problem consists of three labeling tasks - labeling conference function pages, function blocks, and attributes. Different from traditional information extraction tasks, the data in academic conference homepages has complex structural dependencies across multiple Web pages. In addition, there are logical constraints in the data. In this paper, we propose a unified approach, Constrained Hierarchical Conditional Random Fields, to accomplish the three labeling tasks simultaneously. In this approach, complex structural dependencies can be well described. Also, the constrained Viterbi algorithm in the inference process can avoid logical errors. Experimental results on real world conference data have demonstrated that this approach performs better than cascaded labeling methods by 3.6% in F1-measure and that the constrained inference process can improve the accuracy by 14.3%. Based on the proposed approach, we develop a prototype system of use-oriented semantic academic conference calendar. The user simply needs to specify what conferences he/she is interested in. Subsequently, the system finds, extracts, and updates the semantic information from the Web, and then builds a calendar automatically for the user. The semantic conference data can be used in other applications, such as finding sponsors and finding experts. The proposed approach can be used in other information extraction tasks as well.

Categories and Subject Descriptors

I.5.1 [Pattern Recognition]: Models – Statistical

General Terms

Algorithms, Experimentation

Keywords

Constrained Hierarchical Conditional Random Fields, Information Extraction, Semantic Conference Information.

CIKM'08, October 26-30, 2008, Napa Valley California, CA, USA.

1. INTRODUCTION

The semantic information of academic conferences consists of conference details, such as conference names, paper submission deadlines, sponsors, etc. It plays an important role in academic social networks. Understanding such information by machines can bring interesting applications in the Semantic Web. For example, a user-oriented conference calendar can be built to automatically obtain conference information on the Web according to a specific conference list users are interested in. Also, from the information about sponsors and topics of different conferences over years, we can know the history of research interests of certain companies, which can be used to find sponsors for new conferences or to predict the development directions of the companies. Furthermore, memberships of program committees and topics of a conference can be used to help find paper reviewers or experts in certain areas.

Unfortunately, conference information understanding is still an unsolved issue. Previous work has extracted some attributes from call-for-paper (CFP) texts. This approach suffers from the following disadvantages: 1) It is not always easy to find CFP text for every conference. Even though some Websites such as "DB-world" (http://www.cs.wisc.edu/dbworld) provide conference CFPs, they usually cover only conferences of interest of the group. For instance, we could find textual CFPs for only 40% of the top 293 computer science conferences listed at "citeseer" (http://citeseer.ist.psu.edu/impact.html). 2) Plain texts of CFP lose the format and structural information, which can highly improve information extraction result. 3) Not all the conference information is contained in CFP. In our statistics about 1000 conference CFP pages, less than 10% provide sponsor information; in addition, updated deadlines can not be timely reflected in CFP documents.

In general, conference information is provided in conference homepages, which leads us to extract information directly from Websites. Based on our statistics about 293 conferences held from 2004 to 2008, over 96% of conferences have homepages, providing necessary information. All homepages contain format and structural information. Compared with previous Web data extraction tasks, extracting information from conference homepages has two new features: 1) Strong structural dependencies exist across multiple Web pages. For instance, program committee members are listed in the program committee block, and this block usually appears in the program committee page and sometimes also in the call-for-paper page. Figure 1 shows the ontology we have defined for conference homepages. Conference attributes are distributed in different function blocks

and these blocks are distributed in different function pages 2) Some logical constraints exist in conference information. For example, paper deadlines should be earlier than conference dates. Also, deadlines are usually in dates block, rather than others like topics block..

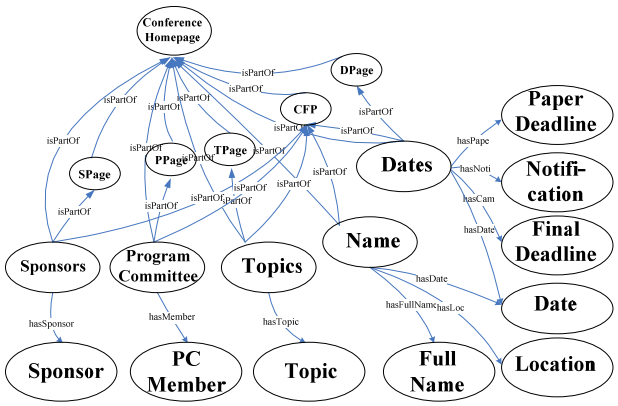

Figure 1. Ontology of conference homepage.

Two questions arise for academic conference homepage understanding: 1) How to describe the complex dependencies among multiple Web pages to help label different attributes in different function blocks from function pages? 2) How to make the inference results satisfy the logical constraints?

This paper addresses these two problems by proposing a Constrained Hierarchical Conditional Random Fields method. The contributions include:

1) We propose a new unified model - Constrained Hierarchical Conditional Random Fields (CHCRF) - to label conference homepage information. This approach combines the ideas of Hierarchical Conditional Random Fields (HCRF) [29] and linear constrained Viterbi inference algorithm [14]. On one hand, it represents the hierarchical structure as probabilistic graph; on the other hand, it expands linear constrained Viterbi into a hierarchical structure, and uses the structure in the inference process. The model can label the three tasks of function pages labeling, function block labeling and attribute labeling simultaneously based on complex hierarchical dependencies among multiple pages, while satisfying the logical constraints.

2) Experimental results in real world conference data have demonstrated that simultaneously labeling the three tasks with the help of hierarchical dependencies to each other performs better than cascaded labeling using linear Conditional Random Fields (CRF) or Support Vector Machine (SVM) methods by 3.6% in F1-measure. At the same time, constrained inference process can avoid logical error, thus improve the accuracy by 14.3% in F1-measure.

3) Based on our approach, we design and implement an interesting Semantic Web application prototype system-Semantic Conference Calendar. In contrast to previous work that manually creates a list of upcoming/current and past conferences, to build a calendar using our system, the user simply needs to specify what conferences he/she is interested in. The system finds, extracts, and updates the semantic information from the Web. We firstly train a SVM classifier to identify conference homepage URLs from search engine (Google) results, given its name and year as keywords, and then employ our approach to label conference information. The semantic data obtained by this system can be used in other applications such as finding sponsors, etc.

The rest of this paper is organized as follows: In section 2, we introduce related work. In section 3, we formalize the problem of academic conference homepage understanding. In section 4 we describe our approach to the problem and in section 5 we give the experimental results. Our design and implementation of the semantic conference calendar application is presented in sections 6. Finally, we summarize this paper in section 7.

2. RELATED WORK

Previous works of extracting semantic conference data are mainly from CFP texts. [16] used rule-based method to extract date and country in conference CFP dataset. [24] employed linear CRF model to extract seven attributes (e.g. title, deadline, location) in another conference CFP dataset with average F1-measure of 66.4%. Pascal Challenge 2003 provided a common platform for researchers to empirically assess methods and techniques devised for information extraction from workshop CFPs [13]. Participants employed different methods to extract 11 attributes for each workshop. For example, [3] used LP^2 algorithm; [8] used CRF model; [17] used SVM classifiers. The best result is 69.8% in average F1-measure. As discussed above, the limited CFP source, ignoring of format and structure, and lack of sponsor and updated information, have led it hard to obtain comprehensive and dynamic conference information from CFP texts. In our work, we obtain conference information directly from Web pages, whose information is much richer.

Many information extraction methods have been proposed. LP^2 [3], Hidden Markov Model (HMM) [10], Maximum Entropy Markov Model (MEMM) [19], linear Conditional Random Field (CRF) [11] [15], Support Vector Machines (SVM) [6], and Voted Perceptron [5] are widely used information extraction models. Some of the methods only model the distribution of contexts of target instances and do not model dependencies between the instances, for example, SVM and Voted Perceptron. Some other methods can model the linear-chain dependencies, for example, HMM, MEMM, and linear CRF. In our work, we employ LP^2, SVM, and linear CRF as our baseline methods since these are most widely used in traditional information extraction tasks.

Conditional Random Fields is a state-of-the-art probabilistic model for information extraction. It is first proposed by [15] for segment and labeling sequences data. Due to effectively utilizing dependencies of elements, it is widely used and developed such as Multi-scale CRF [12], Semi-CRF [23], 2D-CRF [28], TCRF [26], HCRF [29]. In this work, we have implemented a Hierarchical Conditional Random Fields (HCRF) tool, while combining constrained Viterbi algorithm in inference.

Constraints exist in many labeling tasks, adding which into the model will improve labeling results. Kristjansson proposed linear constrained Conditional Random Fields in [14]. By using a constrained Viterbi decoding in the reference process, the optimal fields assignment, which is consistent with some fields explicitly specified or corrected by the user, can be found. Both assignment and features can be constrained in a local way. However, constraints can not be relation of two distant tokens. [22]

proposed Integer Linear Programming Inference process for linear CRF, in which distant constraints of nodes can be dealt with. In [22], constraints were relaxed to ones that can be described in a linear constraint equation with the assignment of each node as variants. However, features cannot be constrained in this method. In our approach, we have expanded constrained Viterbi decoding from linear structure to hierarchical structure and shown how to describe three kinds of constraints in this application.

Several research efforts have been made so far for providing semantic calendar services. For example, [20] developed the RETSINA Calendar Agent (RCAL), which could collect event information from schedules like conference programs published on the Semantic Web. [1] developed OntoWiki. The system provides different views on knowledge database. Calendar is one of the major modules in their system. [9] implemented a system called "e-Wallet" aiming at providing Semantic Web Services including calendar. The main differences of their works from us are that their systems are based on existing semantic conference data. However, there is not much of semantic data on the Web. On the contrary, our system focuses on using information extraction techniques to generate semantic data automatically from the Web.

3. PROBLEM DEFINITION

3.1 Data Representation

Previous works in Web data extraction [29] have shown that vision-tree is a reasonable representation for Web page understanding. Version-tree is generated by a Vision-based Page Segmentation (VIPS) algorithm [2], which utilizes format and structure information in html file to partition Web pages into blocks. In a version-tree, inner nodes represent data blocks, leaf nodes represent atomic units (e.g. element), and root represents the whole page.

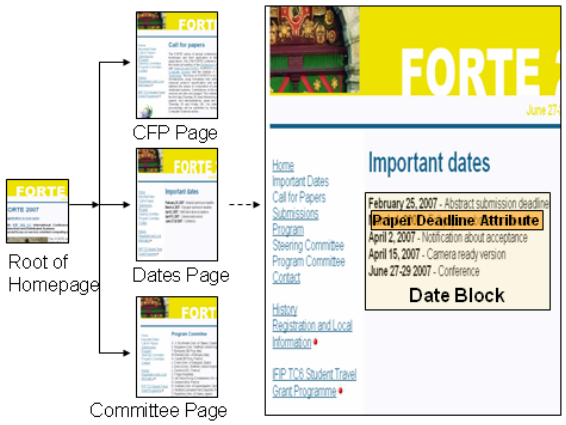

Figure 2. Multiple pages structure in Conf. homepage.

For conference data, different from previous Web data extraction works, information is distributed in multiple pages rather than a single page, and structure dependencies occur across these pages. Figure 2 gives an example of conference root homepage and its linked function pages. To describe such kind of complex dependence information, we propose to combine version-trees of different pages together with the root page. Figure 3 gives conference data representation in a combined version-tree. Here, triangles denote function page nodes, rectangles denote inner data block nodes, and ellipses denote leaf nodes. Blocks denoted by dotted are not fully expanded. We do not expand links in function pages. Our statistic study shows less than 5% information is in these pages.

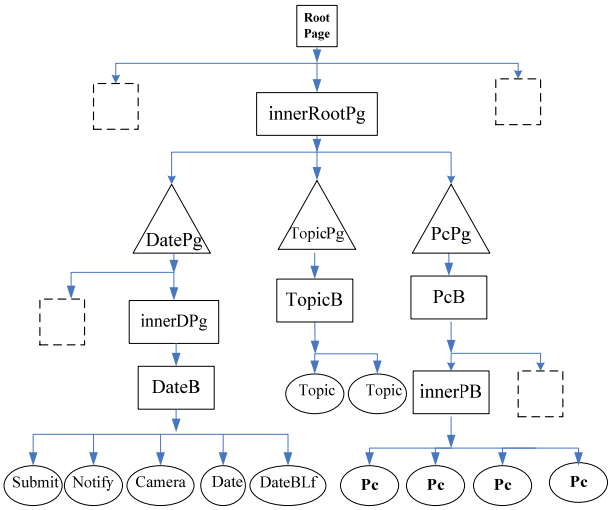

Figure 3. Conf. representation in combined version-tree.

3.2 Three Sub Labeling Tasks

To understand academic conference homepage, three sub tasks are defined based on combined version-trees: 1) Function pages labeling: As many function pages exist in conference Websites, this task is to label these pages in the root page with predefined classes. 2) Function blocks labeling: In conference Web page, detailed information usually occurs in a structured data block. For example, different deadlines occur in "Important Dates" block. It is natural to label these blocks, and meanwhile, these blocks will bring more structure dependence information to the model, improving the results. 3) Attributes labeling: Attributes labeling is labeling the elements in the Web page. The label spaces of these three kinds of nodes are shown in Table 1.

Table 1. Label spaces for different node types

Type	Label Space
Pages	DatePg, PcPg, TopicPg, OtherPg, SponPg, CfpPg
Blocks	TopicB, PcB, DateB, NameB, SponsorB, InnerNB, InnerDB, InnerPB, InnerTB, InnerSB ,InnerRoot, InnerCfpPg, InnerTPg, InnerPPg, InnerDPg, InnerOtherPg, InnerSPg
Attributes	FullName, Location, Date, Submit, Notify, Camera, Sponsor, Topic, Pc, NameBLf, DateBLf, CfpPgLf, TopicPgLf, SponsorPgLf, PcPgLf, DatePgLf, OtherPgLf, RootLf

3.3 Constraints in Conference Data

According to analysis of conference homepage data, three kinds of logical constraints among the information are defined.

1) Label space constraints: The label space of current node is constrained with the label of its parent. For example, if parent's label is "DatePg", the label space of current node is restricted to: if it is a block node, the label can be "InnerDPg", "NameB", "DateB", other labels like "TopicB", "PcB" should not be chosen;

or if it is a leaf node, the label can be "DatePgLf", "Date", and others like "Location" or "Pc" can not be chosen. By doing this, it can enhance the dependence of structure from multiple pages into the model. In our work, this kind of constraints includes all parent-child relations generated from Figure 1.

2) Label occurrence frequency constraints: In a combined version-tree, some information can only occur once. For example, there is only one call-for-paper page in a conference homepage; or, in a "DateB" block, "Submit" only occurs once. In our definition, all five function pages ("DatePg", "PcPg", "TopicPg", "SponPg", "CfpPg".) can occur once at most in one conference homepage, and all four dates information ("Submit", "Notify", "Camera", "Date") can occur once at most in one "DateB".

3) Temporal constraints: There are four attributes related to date information: "Submit", "Notify", "Camera", and "Date". Logically, paper submission deadline is before notification date (expressed as "Submit"< "Notify"). In the same way, "Notify" < "Camera" < "Date".

3.4 Problem Definition

Based on analysis above, academic conference homepage understanding can be described as labeling three kinds of nodes in combined version-tree with different labeling spaces, while making the inference satisfy the logical constraints. We can use a unified probabilistic model to solve the three tasks above. Formally, referring to [29], we define the problem as:

Given a combined version-tree of conference data, let $x=\{x_1, x_2, \ldots, x_n\}$ be the observations of nodes, and let $y=\{y_1, y_2, \ldots, y_n\}$ be possible corresponding labels. The goal is to compute maximum a posteriori (MAP) probability of y, which satisfies the logical constraints, and extract the assignment y^*:

$$y^* = \arg\max p(y \mid x), y \in C, C = \{y \mid y \text{ satisfies constraints}\}$$

Relevant to this problem, [29] has proposed a HCRF to describe complex dependencies among Web page, though our case is more complex as information is distributed in multiple level pages rather than one page. However, no constraints are added in this model. [14] has proposed linear constrained CRF to add constraints into the model, while it can not be used directly in a hierarchical structure. In our model, we propose to combine them together to a Constrained Hierarchical Conditional Random Fields model. We build a HCRF model to describe the complex dependence; in addition, we combine it with hierarchical structured constrained Viterbi decoding in inference to make inference results satisfy the logical constraints, which is expanded from linear constrained Viterbi decoding. It will be described in detail in next section, together with how to employ it in academic conference homepage understanding.

4. CONSTRAINED HIERARCHICAL CONDITIONAL RANDOM FIELDS

In this section, we first introduce how we implement Hierarchical Conditional Random Fields. It is first proposed by [29] and we build a HCRF tool. Then, we explain how to expand linear constrained Viterbi decoding into a hierarchical structured constrained Viterbi decoding, and how to implement it to describe constraints in academic conference homepage understanding. Finally, we give the features defined.

4.1 Hierarchical Conditional Random Fields

Linear Conditional Random Fields is a conditional probability distribution of a sequence of labels given a sequence of observations, represented as $P(Y|X)$, where X denotes the observation sequence and Y the label sequence [15]. The conditional probability is formulized as:

$$p(Y \mid X) = \frac{1}{Z(X)} \exp\left(\lambda \cdot F\left(Y, X\right)\right)$$

where $F(Y, X)$ is feature function vectors defined on cliques of vertices and edges in the linear graph; parameter λ is feature function weights vector corresponding to each feature function, and is to be estimated from the training data; $Z(X)$ is the normalization factor. Like linear CRF, HCRF is a conditional probability distribution of the set of labels given the set of observations in a hierarchical graph, represented also as $P(Y|X)$ [29]. The probabilistic graphical model of HCRF is a hierarchical graph, which is shown in Figure 4 (left). In this graph, there are three kinds of cliques as vertices, edges, and triangles. Expanding cliques with triangles, conditional global probability can still be formulized as the formula above.

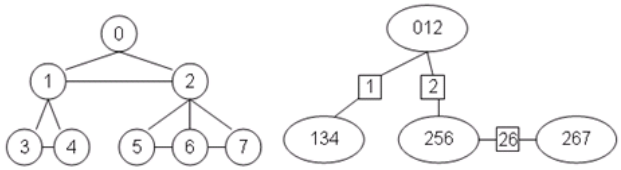

Figure 4. Probabilistic graphical model of HCRF.

Parameters learning problem is to calculate parameter λ by maximizing the log-likelihood from training data $D=\{(x_k,y_k)\}_{k=0}^{N}$.

The optimizing objective function can be written in:

$$L_\lambda = \sum_{k=0}^{N} log\ p_\lambda\left(y_k \mid x_k\right) = \sum_{k}^{N} \left[\lambda \cdot F\left(y_k, x_k\right) - log\ Z_\lambda\left(x_k\right)\right]$$

The gradient of the objective function is:

$$\nabla L_\lambda = \sum_{k=0}^{N} \left[F\left(y_k, x_k\right) - E_{p_\lambda(Y|x_k)} F\left(Y, x_k\right)\right] = \sum_{k=0}^{N} \left[F\left(y_k, x_k\right) - \sum_{y} p_\lambda\left(y \mid x_k\right) F\left(y, x_k\right)\right]$$

In hierarchical graph, according to [7], for one training sample, we should first convert the graph to a junction tree, shown in Figure 4 (right). Then, the expectation part can be calculated in:

$$\sum_{y} p_\lambda\left(y \mid x_k\right) F\left(y, x_k\right) = \sum_{i} \sum_{c_i} p_i\left(c_i \mid x_k\right) \phi_i\left(c_i, x_k\right)$$
$$= \sum_{i} \sum_{y_{i1}, y_{i2}, y_{i3}} p_i\left(y_{i1}, y_{i2}, y_{i3} \mid x_k\right) \phi_i\left(y_{i1}, y_{i2}, y_{i3}, x_k\right)$$

where i denotes clique index, $\phi_i(c_i,x_k)$ denotes clique energy function value corresponding to clique i, including vertex function value, edge function value, and triangle function value. Here, clique i is consisted of node i_1, i_2, and i_3. We employ Belief Propagation [27] to calculate marginal probabilities. For a given x_k, "message" from clique i to j is defined as:

$$m_{ij}\left(c_j, x_k\right) = \sum_{c_i} \phi_i\left(c_i, x_k\right) \varphi_{ij}\left(c_i, c_j, x_k\right) \prod_{I \in N(i) \setminus j} m_{li}\left(c_i, x_k\right)$$

$$\text{where } \varphi_{ij}\left(c_i, c_j, x_k\right) = 1$$

Then,

$$p_i(c_i \mid x_k) = w\phi_i(c_i, x_k) \prod_{j \in N(i)} m_{ji}(c_i, x_k)$$

$$Z_\lambda(x_k) = \sum_{c_i} \phi_i(c_i, x_k) \prod_{j \in N(i)} m_{ji}(c_i, x_k) \quad (c_i \text{ can be any clique})$$

where w is the normalization factor, m_{ij} is message transferred in the graph, and $N(i)$ denotes the neighbors of clique i. Finally, to reduce over fitting, we define a spherical Gaussian weight prior over parameters, and penalize log-likelihood object function as:

$$L_\lambda = \sum_{k=0}^{N} log\ p_\lambda(y_k \mid x_k) - \frac{\|\lambda\|^2}{2\sigma^2} + const$$

with gradient:

$$\nabla L_\lambda = \sum_{k=0}^{N} \left[F(y_k, x_k) - \sum_y p_\lambda(y \mid x_k) F(y, x_k) \right] - \frac{\lambda}{\sigma^2}$$

where const is a constant. We used gradient-based L-BFGS [18], which has previous outperformed other optimization algorithms for linear CRF [25].

4.2 Constrained Inference Process

Kristjansson et al. [14] have shown how to use constrained Viterbi algorithm in linear CRF. We extend this algorithm into hierarchical structure and we show how three kinds of constraints can be solved in this expanded constrained Viterbi algorithm.

Viterbi inference process of junction tree is to maximum messages sending from leaf to root.

$$m_{ij}^*(c_j, x_k) = \underset{c_i}{MAX}\, \phi_i(c_i, x_k) \varphi_{ij}(c_i, c_j, x_k) \prod_{l \in N(i) \setminus j} m_{li}^*(c_i, x_k)$$

Here, $\phi_i(c_i,x_k)$ can be calculated by feature function, $\phi_{ij}(c_i,c_j,x_k)$=1. In hierarchical structured constrained Viterbi algorithm, both these energy function can be set zero (or a very small number) if assignment encounters constraints. It can be formulized as:

$$m_{ij}^*(c_j, x_k) = \begin{cases} \underset{c_i}{MAX}\, \phi_i(c_i, x_k) \varphi_{ij}(c_i, c_j, x_k) \prod_{l \in N(i) \setminus j} m_{li}^*(c_i, x_k) & \textit{fit constraints} \\ 0 & \textit{not fit constraints} \end{cases}$$

In this way, the inference result will be an assignment fitting constraints with the largest probability.

The three types of constraints above can be described as follows.

1) Label space constraints: if the assignment in one clique disobeys label space constraints, $\phi_i(c_i,x_k)$ will just be set zero to avoid assignment. If for every clique, parent and child satisfy the constraints, in the whole graph, it must also satisfy these constraints.

2) Label occurrence frequency constraints: In Viterbi algorithm, assignment of one clique is only determined by its neighbors, so it is hard to constrain the occurrence frequency of a label in one Viterbi process. However, we can simply use two-step inference iterations to solve this problem. Consider "CfpPg" for example, in the first round, we run a Viterbi algorithm to find the most likely "CfpPg" and tag the node. And in the second round, we add a local constraint that tagged node is the one and the only one having the label of "CfpPg". We can find the most likely "CfpPg" by comparing the value of marginal probability of cliques containing "CfpPg" label, calculated again by Belief Propagation.

3) Temporal constraints: Date information elements usually occur together in conference pages. So in two adjacent cliques with three continuous elements in "DateB", it can cover two date attributes in most cases. In this way, if date attributes in one clique disobey the time order, $\phi_i(c_i,x_k)$ will be set zero; if date attributes across two adjacent cliques disobey, $\phi_{ij}(c_i,c_j,x_k)$ will be set zero. We define some rules to recognize whether current element contains date information and convert it into a format like "12/15/2008" for comparing.

4.3 Features

Features used in academic conference homepage understanding include features of elements, features of inner blocks, and features of pages.

4.3.1 Element Features

Word Features: Include words in current element and words in context elements. The window size is 1.

Morphology Features: The morphology of an element is divided into capitalized short terms, normal short terms, capitalized short sentence, normal short sentence, and paragraph.

Date Feature: Whether current element contains dates information. We use rules to identify dates information such as whether it contains keywords like "June", "2008", etc.

Location Feature: Whether current element contains location information. We have built a location list from http://www.world-gazetteer.com/dataen.zip. If a word in current element matches one item in the list, it is recognized as location information.

Title Keywords Features: Whether current element contains keywords occurring frequently in titles like "st", "nd", "rd", "th", "conference", "workshop", "symposium", etc.

First Location Feature: The first location in the root page. Usually, the first location in the root page is likely to be the location of the conference, referring to [16].

Latest Date Feature: Whether current element contains latest date among the homepage. Referring to [16], latest date in a conference homepage is likely to be the date of the conference. We use some rules to identify date information and compare them between each other.

4.3.2 Block Features

Children Number Feature: Number of children of the block.

Position Features: Center position of current block. The center can be calculated from features given by VIPS [2].

Area Features: Area of current block. The Area can be calculated from version-tree features given by VIPS [2].

Block Keywords Features: Whether the first element before this block contains keywords like "Topics", "Program Committee", etc. This can help to recognize "function blocks".

4.3.3 Page Features

Page Keywords Features: "Function pages" usually have some hint words in the hyperlink. Here, we define "topic", "scale", "scope", "theme", etc, for "TopicPg"; "important", "key", "date",

"deadline", etc, for "DatePg"; "call", "paper", etc, for "CfpPg"; "PC", "officer", "committee", "organization", etc, for "PcPg"; "sponsor", etc, for "SponPg".

5. EXPERIMENTS

5.1 Experiments Setup

5.1.1 Datasets

In total, we collected 570 conference homepages of 293 conferences during 2004-2008 containing the information defined in section 3, from top ranking conferences announced in "Citeseer" (http://citeseer.ist.psu.edu/impact.html). For each conference sample, we downloaded the root page and all linked pages in it. Some rules were defined to remove some linked pages which were sure not to be "function pages" to reduce the size of sample. Then, VIPS [2] was used to convert each page into a version-tree file, and then we jointed them into a combined version-tree. Human annotators conducted annotation on the combined version-trees. A spec was created to guide annotation process. All "function pages", "function blocks", and "attributes" defined in Figure 1 were labeled. For disagreements in the annotation, we conducted "majority voting". Table 2 shows the statistic of our dataset. We used 3/4 of them for training and others for testing. Four-fold cross-validation was used in experiments.

5.1.2 Evaluation Measures

For all three labeling tasks, we used standard precision, recall and F1-measure (for definition of measures, see [21]) to evaluate experimental results. In addition, we used average F1-measure to evaluate different methods. In particular, a block is considered as correctly labeled if it contains all the attributes in it while being tolerant for one layer difference.

5.1.3 Experiments Design

To compare our model with traditional rule-based information extraction methods, we firstly employ LP^2 as our baseline, which is the best method in previous works of conference information extraction in Pascal Challenge [13]. The algorithm tool we use is from Amilcare [4].

To evaluate our model's effectiveness of incorporating more hierarchical dependencies for labeling, we chose two cascaded labeling methods, which firstly label "function pages", based on the results, then label "function blocks", and finally, label "attributes". One is SVM method, the other is Linear Conditional Random Fields (LCRF) method. Both methods are widely used in previous information extraction tasks. In these two methods, we converted Web page into a sequence while remaining their format and structure information as features. For SVM method, we trained a classifier for each label; for LCRF, we trained one model for all the labels. We used existing SVM tool (http://svmlight.joachims.org) and CRF tool (http://crfpp.sourceforge.net) to do these experiments.

To evaluate our model's effectiveness of utilizing the constraints among conference labels in inference, we also used standard Viterbi algorithms in inference (HCRF) as baseline.

We have developed HCRF and CHCRF tools. Both HCRF and CHCRF experiments were done using our tools. As LP^2 and LCRF are difficult to use in "function blocks labeling", we only did experiments with them in other two tasks.

Table 2. Statistic on conference dataset

Label Items		Occurrence	
		#Instance	Percentage of containing in homepages
Pages	DatePg	163	35.3%
	CfpPg	297	64.1%
	TopicPg	43	9.3%
	PcPg	320	69.1%
	SponPg	107	23.1%
Blocks	DateB	597	89.4%
	NameB	405	70.2%
	TopicB	540	77.9%
	PcB	608	88.4%
	SponB	442	50.0%
Attributes	Submit	763	89.4%
	Notify	602	87.5%
	Camera	505	77.9%
	Date	559	74.0%
	Location	434	74.0%
	FullName	730	76.9%
	Topic	7622	82.7%
	Pc	3281	90.4%
	Sponsor	3586	58.7%

5.2 Experimental Results

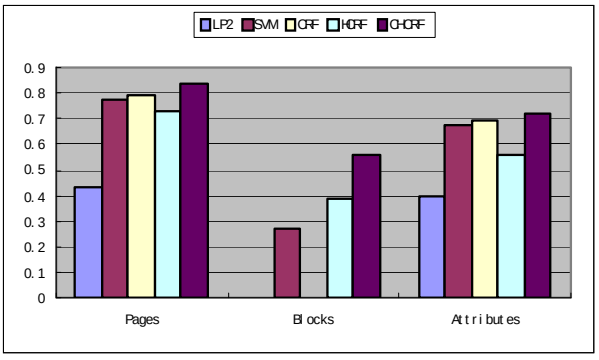

Figure 5. Average F1-measure of different methods.

Table 3. Evaluation Results of different methods for "function pages", "function blocks", and "attributes" labeling (%)

Methods		LP^2			SVM			LCRF			HCRF			CHCRF		
Measure		Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
pages	Date	56.3	34.6	42.9	84.4	83.7	80.4	84.0	84.6	84.3	95.3	69.1	80.1	93.5	86.0	**89.6**
	CFP	56.9	46.8	51.4	73.4	85.7	79.1	75.3	82.4	78.7	56.3	95.4	70.8	85.9	96.1	**90.7**
	Topic	45.9	17.2	25.0	64.2	95.3	76.7	79.2	87.4	83.1	92.0	49.6	64.5	92.3	80.0	**85.7**
	PC	57.0	48.6	52.5	68.8	84.9	76.0	71.0	81.4	75.8	78.3	72.1	75.1	94.5	88.5	**91.4**
	Spon	61.2	34.2	43.9	63.7	83.2	72.2	65.7	86.3	**74.6**	73.4	75.2	74.3	58.6	63.0	60.7
blocks	Date	-	-	-	35.3	63.1	45.3	-	-	-	46.1	58.6	51.6	63.0	72.8	**67.5**
	Title	-	-	-	12.0	9.0	10.3	-	-	-	31.7	42.2	36.2	50.7	62.1	**55.8**
	Topic	-	-	-	22.0	51.7	30.9	-	-	-	42.7	61.2	50.3	58.5	75.6	**66.6**
	PC	-	-	-	21.3	64.3	32.0	-	-	-	47.4	48.7	48.0	64.9	67.4	**66.1**
	Spon	-	-	-	19.4	15.7	17.4	-	-	-	6.9	14.7	9.4	17.9	29.4	**22.3**
attri-butes	Subm	34.0	27.0	30.1	60.0	70.3	**64.7**	62.5	63.7	63.1	38.9	66.5	49.1	66.3	51.5	58.0
	Noti	40.0	47.0	43.2	70.7	82.0	75.9	75.0	72.0	73.5	86.2	72.4	78.7	85.9	77.8	**81.6**
	Came	50.0	27.0	35.1	66.9	75.1	70.8	86.8	60.9	71.6	70.1	71.4	70.7	79.5	82.7	**81.1**
	Date	75.0	42.0	53.8	54.3	75.0	63.0	71.5	66.9	**69.1**	42.5	61.7	50.3	73.1	57.7	64.5
	Loca	33.0	10.0	15.3	73.6	61.7	67.1	90.0	52.6	66.4	70.0	45.6	55.2	75.3	66.0	**70.3**
	Title	71.0	22.0	33.6	65.7	66.9	66.3	69.4	62.2	65.6	35.5	46.8	40.4	69.4	72.2	**70.8**
	Topic	47.0	45.0	50.0	77.0	79.3	78.1	87.8	72.3	79.3	34.8	83.0	49.0	86.6	83.7	**85.1**
	PC	41.0	52.0	45.8	67.3	89.5	76.8	78.0	86.0	81.8	87.1	62.3	72.6	85.2	90.1	**87.6**
	Spon	58.0	53.0	**55.4**	36.3	70.4	47.9	42.0	71.1	52.8	33.4	47.4	39.2	43.8	57.1	49.6

Table 3 shows the four-fold cross-validation results for all three labeling tasks in different methods. And Figure 5 shows the average F1-measure using different methods.

From the results we can see that our proposed CHCRF can achieve the best result on average in all three labeling tasks. Based on F1-measure, in "function pages labeling", CHCRF outperforms LP^2 by 40.4%, SVM by 6.0%, LCRF by 4.3%, and HCRF by 10.7%; in "function blocks labeling", CHCRF outperforms SVM by 28.4%, and HCRF by 16.4%; in "attributes labeling", CHCRF outperforms LP^2 by 32.2%, SVM by 4.2%, LCRF by 2.8%, and HCRF by 15.9%. In total, CHCRF outperforms cascaded method (LCRF) by 3.6% (F1-measure) and non-constrained HCRF method by 14.3% (F1-measure).

5.3 Discussions

5.3.1 Effectiveness of hierarchical structure

By comparing the results of CHCRF with the ones from SVM and LCRF, we can see simultaneously labeling the three tasks has received better results than labeling them in a cascaded way. It is mainly because CHCRF takes more structure information into the model, which helps to utilize complex dependencies of all the information. Based on this, three sub labeling tasks can help each other, while cascaded methods propagate errors in each step.

5.3.2 Effectiveness of constrained inference

Our CHCRF outperforms HCRF, and the main reason is the constrained inference. The constrained Viterbi decoding makes inference fit all logical constraints defined, so the accuracy has been improved. We found that HCRF performed worse than cascaded methods like SVM and LCRF. There are two reasons: One is that without constrained inference, the structure dependencies can not be correctly described. Sometimes, "Pc" occurs in "TopicB", or "Location" occurs in "PcB". Without correctly describing the distant dependencies, HCRF lost its advantages. The other reason is that data representation is simpler in cascaded methods. In cascaded methods, there is no inner node. All the sequence units are leaf nodes in the version-tree. The representation in a version-tree has almost twice size as the sequence, making the labeling more complex.

5.3.3 LCRF performing better than SVM

From experimental results, LCRF performed better than SVM. That is because their data representations are in the same size, and LCRF can utilize label dependencies of adjacent contexts. This helped to improve labeling results.

5.3.4 Analysis for LP^2

Traditional rule-based LP^2 method did not receive good performance in this Web page labeling task. That is because rule-based methods mainly depend on contexts hints. In this task, structure and label dependences information are helpful. The LP^2 algorithm, however, can not use structure information as features and can not take advantages of label dependences. Also, it can not utilize effective features defined manually. In "location" labeling, it can not recognize a location through a list, which leads to the low accuracy.

5.3.5 Error analysis for sponsor information

We can see from the results in all methods, sponsor information labeling has received the worst accuracy. This is mainly because sponsor is usually presented with pictures in Web page rather than texts. In our building of combined version-tree, we replaced the

images by the “alt” attributes in the element. 74% errors came from that “alt” attribute has no hint word for the sponsor or there were no “alt” attribute in the elements so there were no hints to recognize the content. Others are from errors of model.

6. SEMANTIC CONFERENCE CALENDAR

6.1 Background

As we mentioned in our first section, semantic conference information can bring many fascinating Semantic Web applications. Based on the semantic conference data obtained using our proposed Constrained Hierarchical Conditional Random Fields, we design and implement a prototype system of semantic conference calendar.

A conference calendar is typically known as a list of upcoming/current and past conferences. It also includes some important information related to the conferences, for example, conference date, conference full name, conference location, conference scope, paper submission date, and paper notification date. Such information is very useful for both academic and industrial researchers in their schedule decisions.

Traditionally, conference calendar is viewed as an engineering issue and is constructed manually. For example, Association for Information Systems (http://www.isworld.org) provides a “Conference CFP Page” that contains information about call-for-paper for conferences of interest to the global Information System community. DB-world (http://www.cs.wisc.edu/dbworld) provides a comprehensive and frequently updated list of events such as conferences and workshops in Computer Science (An ACM SIGMOD resource). Disadvantages of manual style are obvious: 1) Semantic information is limited. Much useful information is presented in the unstructured plain text; 2) It is not easy to implement a customizable conference calendar. Most of users would not like to see the comprehensive list as AIS and DB-world, as they might only be interested in a very small subset of the conferences.

In our work, we describe a Semantic Web application that builds a customizable conference calendar. In contrast to previous works aiming at manually providing a comprehensive list of upcoming/current and past conferences, in this work we engage in implementing a semantic conference calendar which can achieve to automatically get information from the Web using conference homepage understanding proposed in this paper. In this system, to build one’s calendar, the user simply needs to specify what conferences he/she is interested in. The system finds and extracts the semantic information from the Web automatically. The conference information also can be updated (e.g., for different years) automatically from the Web.

6.2 Motivating Example

Our system here aims at providing a user-oriented conference calendar. The scenario is defined as follow:

A user, for instance an academic researcher, wants to be kept reminded of several conferences he/she is interested in. He/she inputs the conference names (acronyms or full name) into our system. The system automatically identifies the homepages of the conferences for every year; then extracts the semantic conference information from conference homepage. The extracted conference information is filled into a calendar and thus a personalized calendar is created. Figure 6 shows an example of conference calendar. The left-top part is the homepage identified by our system for ISWC’2007; the right part is the extraction process; and the left-bottom part is the constructed calendar.

6.3 Our Solution

Consequently, three problems need to be solved to build an automatic calendar: 1) how to find the homepage of a conference; 2) how to label the conference data from the pages; and 3) how to integrate the labeled data into our system.

For the second problem, we have proposed a Constrained Hierarchical Conditional Random Fields approach. The details are presented in previous sections. In this section, we focus on the other two problems.

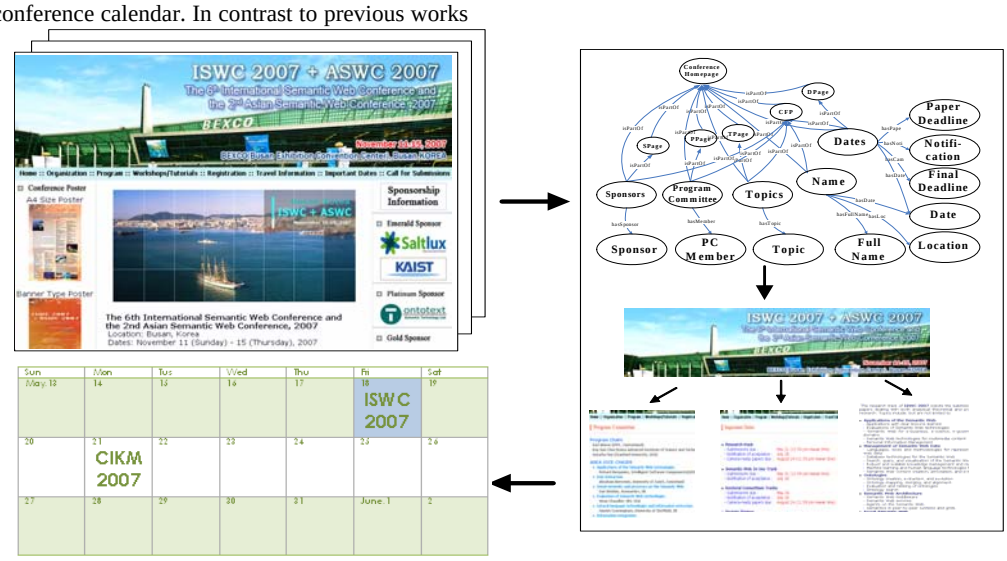

Figure 6. Semantic Conference Calendar.

6.3.1 Conference Homepage Identification

Finding relevant conference homepage is a precondition of obtaining conference information. Naturally, existing search engine provides a good way to find conference homepages. We use Google as our search engine. Our statistics on 293 conferences during 2004 to 2008 with input of the acronym of conference have shown that over 96% conference homepages can be returned in the first 10 items by Google, and 69.3% can be returned in the first item. Then, we formalize conference homepage identification problem as a classification problem. Given the first 10 results from search engine based on acronym and year for a conference, the task is to identify which result is the homepage of corresponding conference. Sometimes two academic conferences have the same acronym, and in this case, both need to be identified.

We proposed a SVM method to solve the problem. The relevant SVM algorithms are described in [6]. The process consists of training and identifying. In the training, we trained a classifier from labeled dataset, and effective features were defined to improve the results including:

URL Pattern Features: Whether the URL contains patterns like "kdd 2008", "org", and "index".

Position Features: Whether the URL is the first, first three, or first five results returned by search engine.

Page Title Feature: Whether the page's title has patterns like "CIKM 2008", "CIKM - 2008", "CIKM' 08", etc.

Hyperlink Features: Whether the page contains hyperlinks like "important dates", "program committee", etc.

Our dataset contains 1046 conference homepages. We used the acronym and year as keywords to put into the search engine, the other results returned were seen as negative samples. Half of them were used for training and others for testing. Two-fold cross-validation experiments have shown that our methods can achieve 69.5% in precision, 79.3% in recall, and 74.1% in F1-measure.

6.3.2 Information Integration

In information integration, we normalize the labeled information. The task is to fill a template in the database. Specifically, 1) When the system finds more than one instances of an attribute, and the values are different, we select the one that has the highest likelihood as the value of the attribute; 2) When the value of the instances are the same, but with different representations (e.g., "June 10, 2007" and "06/10/2007"), we normalize the representations (e.g. both "June 10, 2007" and "06/10/2007" are normalized as "2007-06-10"), and store them in the database.

6.4 Implementation

Our calendar system targets at providing personalized services for users. The services include:

1) Personalized Calendar. The user selects/inputs conference names and the system creates a personalized calendar automatically and keeps reminds of the user.

2) Conference Search. The user input the keywords and the system returns the detailed conference information that is extracted from the Web automatically.

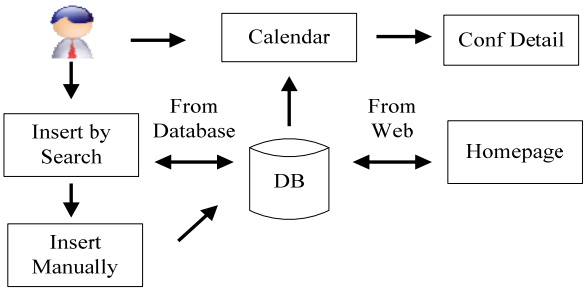

Figure 7. Structure of Calendar System.

The processing of the system is shown in Figure 7. When building the calendar, users can search conference by keywords, the corresponding conference will return. If results do not satisfy the user, in the spirit of Web 2.0, users can insert conference information manually, which will also be kept in database so that other users can share this, too.

The extracted semantic conference data is also very useful for data mining tasks from the social network. Accurate extraction of the conference is essential to expertise conference finding and is greatly helpful for the other mining issues like "sponsors finding", "experts finding", etc.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated the problem of academic conference homepage understanding. Conference information has complex structural dependencies across multiple pages, and has inherent logical constraints. Based on these features, we have proposed a new unified approach, Constrained Hierarchical Conditional Random Fields, to solve the problem by combining the ideas of Hierarchical Conditional Random Fields and Constrained Linear Conditional Random Fields. Experimental results on real world data have demonstrated that this approach performs better than both the cascaded approach and Hierarchical Conditional Random Fields without constraints.

Based on our conference homepage understanding technique, we have designed and implemented a practical Semantic Web application, Semantic Conference Calendar. It can automatically search and extract conference information from the Web and build a calendar for users. The semantic data can also be used in other Web applications such as finding sponsors, predicting company interests, and finding paper reviewers.

As future work, we plan to combine different machine learning methods to improve the accuracy. We also want to build more Semantic Web applications based on the semantic conference data.

8. ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China (90604025, 60703059), Chinese National Key Foundation Research and Development Plan (2007CB310803), and Chinese Young Faculty Research Funding (20070003093). Thank Jun Zhu, Xiaobing Liu and Ali Daud for necessary discussions.

9. REFERENCE

[1] Auer, S., Dietzold, S., and Riechert, T. OntoWiki – A Tool for Social, Semantic Collaboration. In *Proc. of ISWC*, 2006.

[2] Cai, D., Yu, S., Wen, J., and Ma, W. Block-based Web Search. In *Proc. of SIGIR,* 2004, 456-463.

[3] Ciravegna, F. $(LP)^2$ An Adaptive Algorithm for Information Extraction from Web-related Texts. In *Proc. of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, Seattle, USA, 2001.

[4] Ciravegna, F., Dingli, A., Iria, J., and Wilks, Y. Multi-strategy Definition of Annotation Services in Melita, In *Proc. of ISWC'2003 Workshop on Human Language Technology for the Semantic Web and Web Services*, 2003, 97-107.

[5] Collins, M. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proc. of EMNLP,* 2002.

[6] Cortes, C. and Vapnik, V. Support Vector Networks. *Machine Learning*, 1995, 20: 273-297.

[7] Cowell, R., Dawid, A., Lauritzen, S., and Spiegelhalter, D. *Probabilistic Networks and Expert Systems*. Springer-Verlag, New York, 1999.

[8] Cox, C., Nicolson, J., Finkel, J., and Manning, C. Template Sampling for Leveraging Domain Knowledge in Information Extraction. In *PASCAL Challenges,* 2005.

[9] Gandon, F., and Sadeh, N. A Semantic eWallet to Reconcile Privacy and Context Awareness. In *Proc. of ISWC,* 2003.

[10] Ghahramani, Z. and Jordan, M.I. Factorial Hidden Markov Models. *Machine Learning*, 1997, 29: 245-273.

[11] Hammersley, J. and Clifford, P. Markov fields on Finite Graphs and Lattices. 1971.

[12] He, X., Zemel, R., and Carreira-Perpiñán, M. Multiscale Conditional Random Fields for Image Labeling. In *Proc of CVPR,* 2004, 695-702.

[13] Ireson, N., Ciravegna, F., Califf, M.E., Freitag, D., Kushmerick, N., and Lavelli, A. Evaluating Machine Learning for Information Extraction. In *Proc. of the 22nd International Conference on Machine Learning*, 2005, 345–352.

[14] Kristjansson, T., Culotta, A., Viola, P., and McCallum, A. Interactive Information Extraction with Constrained Condition Random Fields. In *Proc of AAAI,*.2004, 412-418.

[15] Lafferty, J., McCallum, A., and Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML,* 2001, 282-289.

[16] Lazarinis, F. Combining Information Retrieval with Information Extraction for Efficient Retrieval of Calls for Papers. In *Proc. of IRSG,* 1998.

[17] Li, Y., Bontcheva, K., and Cunningham, H. Using Uneven Margins SVM and Perceptron for Information Extraction. In *Proc. of CoNLL*, 2005.

[18] Liu, D. and Nocedal, J. On the Limited Memory BFGS Method for Large Scale Optimization. *Mathematical Programming*, 1989, 503-528.

[19] McCallum, A., Freitag, D., and Pereira, F. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proc. of ICML,* 2000, 591-598.

[20] Payne, T., Singh, R., and Sycara, K. Browsing Schedules – an Agent–Based Approach to Navigating the Semantic Web. In *Proc. of ISWC,* 2002, 469-474.

[21] Rijsbergen, C. *Information Retrieval*. 1979.

[22] Roth, D. and Yih, W. Integer Linear Programming Inference for Conditional Random Fields. In *Proc. of ICML,* 2005, 736-743.

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

[24] Schneider, K. Information Extraction from Calls for Papers with Conditional Random Fields and Layout Features. In *Proc. of AICS*, 2005, 267-276.

[25] Sha, F. and Pereira, F. Shallow Parsing with Conditional Random Fields. In *Proc. of HLT-NAACL,* 2003.

[26] Tang, J., Hong, M., Li, J., and Liang, B. Tree-structured Conditional Random Fields for Semantic Annotation. In *Proc. of ISWC*, 2006, 640-653.

[27] Yedidia, J., Freeman, W., and Weiss, Y. Generalized Belief Propagation. In *Proc. of NIPS,* 2000.

[28] Zhu, J., Nie, Z., Wen, J., Zhang, B., and Ma, W. 2D Conditional Random Fields for Web Information Extraction. In *Proc. of ICML,* 2005, 1044-1051.

[29] Zhu, J., Nie, Z., Wen, J., Zhang, B., and Ma, W. Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. *In Proc. of KDD,* 2006, 494-503