

Collaborative Information Extraction and Mining from Multiple Web Documents*

Tak-Lam Wong and Wai Lam and Shing-Kit Chan
Department of Systems Engineering And Engineering Management
The Chinese University of Hong Kong
Hong Kong
{wongtl,wlam,skchan}@se.cuhk.edu.hk

Keywords: text mining, web mining, graphical models

Abstract

We develop an unsupervised framework which can collaboratively extract information from multiple Web pages, as well as conduct feature mining tasks in a *unified* model. Our model allows tight interactions of the two tasks removing the unnecessary boundary between the two tasks. It is beneficial for both tasks since the decisions for information extraction and feature mining can be done in a coherent manner assigning solutions optimizing the quality of both tasks and at the same time eliminating the potential conflicts. Our approach is designed based on an undirected graphical model which can model the inter-dependence between the neighbouring tokens within the same Web page, as well as tokens in different Web pages. Multiple Web pages are considered under this model and the information can be extracted collectively. This design also leads to another characteristic of our framework in that it can conduct mining across Web pages simultaneously. We demonstrate the efficacy of our model by applying it to the important product feature mining application. Extensive experiments on real-world data have been conducted to evaluate our framework.

1 Introduction

The World Wide Web contains enormous amount of Web pages which are accessible by users. This provides a very useful and helpful means of gathering information. However, it is no easy task for users to digest the massive, but poorly organized and formatted information in the Web pages. Though online search engines are helpful, one major shortcoming is the unit of the search results is either a Web site or a Web page. Human effort is still required to identify and digest the highly precise information within different Web pages. Information

*The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Nos: CUHK 4179/03E and CUHK4193/04E), the Direct Grant of the Faculty of Engineering, CUHK (Project Code: 2050363), and CUHK Strategic Grant (No: 4410001). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.



Figure 1: A sample of a Web page about a digital camera collected from a retailer Web site.

extraction techniques aim at automatically extracting precise and useful text fragments from textual documents. Unlike free texts which are largely grammatical or structured texts with rigid format, information extraction from semi-structured documents such as Web pages poses more challenges since they are mixed with portions of weakly grammatical free texts fragments and highly structured mark-up tags. Recently, various approaches using machine learning techniques have been proposed to automatically train an information extraction system from a set of training examples.

One common assumption of existing approaches is that the information contained in the Web pages of different sites is considered independent. Hence the information extraction task is conducted separately without interactions for each page. For example, consider the Web pages shown in Figures 1 and 2. These Web pages are collected from two different Web sites, but they contain the product feature information of the same digital camera. To extract the important product features, existing methods treat the two Web pages separately.



Figure 2: A sample of a Web page about a digital camera collected from a retailer Web site different from the one shown in Figure 1.

However, these two Web pages actually have mutual influence and contain useful clues which can help extraction from each other. For instance, one product feature extracted from Figure 2 is “6.1 megapixel CCD Sensor”. Such extraction strengthens the confidence in extracting the text fragment “6.1-megapixel” in Figure 1 and classifying it as an important feature. From the layout format of Figure 1, we can then infer that the product features are organized in a list format and one can extract more product features such as “SLR”, “1.8” LCD”, etc. Similarly, the extraction results in Figure 1 can again guide the extraction in Figure 2. Consequently, the extraction from Web pages interact with each other and hence the information should be considered collectively.

Recently, various approaches have been proposed to leverage the extracted information for data mining applications. One example is the automatic travel assistance agent which makes use of the automatically extracted data from airline company Web sites to help travelers plan their trips wisely [2]. However, existing approaches suffer from several limitations. First, most existing information extraction approaches treat each potential candidate entity as independent. They do not consider influence or relationships among entities during the extraction process. Second, they typically perform information extraction and mining as two separate independent steps without interactions. Information extraction is first conducted to identify entities from the textual content. Then feature mining or data mining algorithms are applied on the extracted entities to generate or discover certain kinds of relationships among the items. One example is to conduct identity resolution or record linking analysis on extracted entities from text documents. It is obvious that the errors made in the information extraction task will compound with the errors produced in the mining process. Hence it is common to find that the performance of mining is considerably inferior compared with the information extraction performance. Third, existing methods cannot handle

mining across different documents.

Similar to other supervised learning approach, another shortcoming of the many existing information extraction learning techniques is the need for manually provided training examples. A substantial amount of human effort is required to prepare a large set of training examples in advance. This task is obviously tedious, time-consuming, and requires high level of expertise. Moreover, due to the difference between the layout format of the Web pages, a model trained from the Web pages collected from a particular Web site is not able to extract information from Web pages collected from other Web sites. To extract information from other Web sites, one requires a separate human effort to prepare another set of training examples. Therefore, though a collective approach offers some advantages in information extraction from multiple Web pages, such method must cope with the difficulty of different layout format simultaneously.

We propose an unsupervised learning approach which can collaboratively extract information from multiple Web pages, as well as conduct feature mining task in a *unified* model. Our approach allows tight interactions of the two tasks removing the unnecessary boundary between the two tasks. It is beneficial for both tasks since the decisions for information extraction and feature mining can be done in a coherent manner assigning solutions optimizing the quality of both tasks and at the same time eliminating potential conflicts. Our approach is designed based on an undirected graphical model namely Conditional Random Fields [15]. This graph can model the inter-dependence between the neighbouring tokens within the same Web page, as well as tokens in different Web pages. We formulate the problem of information extraction and feature mining as a label assignment problem on this graphical model. Multiple Web pages can be considered using a unified model and the information can be extracted by labeling the tokens collectively in a single framework. This design leads to another characteristic of our framework in that it can conduct mining across Web documents simultaneously.

The layout format of the Web pages contains very useful clues in extraction. For example, the product features in the retailer Web sites are normally organized in a certain regular format such as list or table in the Web pages. Such layout format information can help identify the text fragments to be extracted. However, since Web pages collected from different Web sites are in different layout formats, an expectation-maximization (EM) based adaptive parameter learning method is designed to cope with this difficulty. We also consider the DOM structure of the Web pages. The DOM structure can effectively represent the structure of the

Web page. We develop an information-theoretic method to analyze the DOM structure and find the informative blocks which contain useful text fragments with similar purpose. The identified informative blocks are then utilized in the automatic construction of the graph structure.

Our approach allows easy incorporation of external knowledge sources such as attribute lexicons and extracts the information from Web pages in an unsupervised learning manner. Very often, users already have some prior knowledge in the domain from which the information is extracted. Our method considers two types of prior knowledge of the domain. The first type is a small set of attribute lexicons. For example, we can easily provide a small set of common product attributes for the digital camera domain. In our experiment, we use a lexicon less than 20 attributes such as “resolution”, “pixel”. Note that this lexicon acts as a seed and our framework is able to discover a large amount of previously unseen product features. The second type of prior knowledge is related to the layout format of the Web pages. Useful information is normally formatted in a certain regular layout format and located in nearby position. For example, the product features in Figure 1 are organized in a list format and in the same informative block. Our framework is able to learn the relationship between the product features and the layout format and then automatically discover more useful information.

We demonstrate the efficacy of our model by applying it to the important product feature mining application. Its objective is to extract the product features of the products listed in the retailer Web sites, and identify the *important* product features. Normally, each product has its distinguished and important product features. These important product features are normally displayed with easily perceivable format such as in bolded text, italic, or in some special colors. Discovering these important product features can help users become knowledgeable about the product. We have conducted extensive experiments to demonstrate the effectiveness of our model.

2 Related Work

Our framework is able to extract information and discover useful knowledge from a collection of Web pages from different information sources under a unified model. Various techniques have been proposed to extract information from semi-structured documents such as Web pages [14, 19]. One promising technique is to make use of wrappers [13]. A wrapper normally consists of a set of extraction rules which can identify the precise text fragment from Web documents. Wrapper induction methods apply machine learning technique to

automatically learn the extraction rules from a set of training examples [1, 5, 6, 9, 21]. One major limitation of wrapper induction approaches is that they merely extract information, but cannot discover new patterns. Typically a separate mining task needs to be conducted. The second disadvantage of existing methods is that the learned wrapper can only extract information from the same information source from where the training examples are obtained. For example, if the extraction rules are learned from the training examples collected from a particular Web site, the learned extraction rules are unlikely able to extract information from other Web sites. Another shortcoming is that they can only extract the attributes specified in the training examples. For example, if we just annotate the start time, end time, location, and speaker in the training example in the seminar announcement domain, the learned wrapper can only extract these four attributes. Some other useful information such as the title of the seminar cannot be extracted. Recently, various techniques have been proposed for collectively conducting information extraction and data mining [17]. For example, Wellner et al. proposed an approach for extracting different fields in citation and solving the citation matching problem using conditional random fields [22]. McCallum and Wellner also proposed an approach to extracting proper nouns and linking the extracted proper nouns using a single model [18]. Bunescu and Mooney proposed to use relational Markov networks to collectively extract information from documents [3]. One major difference between these methods and our approach is that our approach is an unsupervised method and does not require any training examples.

We attempted to tackle this problem in our previous work for discovering new attributes in Web pages and reducing the human effort in preparing training examples by adapting the extraction knowledge from one information source to other previously unseen information sources [23]. However, the problem of preparing training examples is still a major limitation in supervised learning. Some techniques have also been developed for fully automatic information extraction from Web pages without using any training examples. For example, IEPAD is a system aiming at extracting information by recognizing the repeated patterns in the Web pages [4]. MDR is a system which mines the data region in a Web page by discovering the repeated pattern in HTML tag trees [16]. Heuristics are then applied to extract the information from the data region. Therefore, both IEPAD and MDR assume that the input Web pages contain multiple records. Another system called ROADRUNNER also makes use of repeated patterns for information extraction [7]. Sometimes, Web pages

of some Web sites are generated by an automatic Web page generation program. Though the content of the Web pages are different, their layout format are similar. The idea of ROADRUNNER is to make use of this evidence and recognize the repeated patterns appeared in the Web pages. However, the Web pages are required to have similar layout format and this may not be true in Web pages collected from different sources. Grenager et al. apply hidden Markov model and exploit prior knowledge to extract information in an unsupervised manner [10]. However, the quality of the extracted data is unlikely suitable for subsequent data mining tasks.

We have also applied our framework to the Web mining application namely important feature mining. Some existing methods related to product feature mining have been developed. For example, Hu and Liu [11] proposed a system aiming at summarizing customer reviews on a product posted on the Web sites. They aim at classifying sentences with subjective orientation and make use of subjective words such as “good”, “perfect” as the clues. To accomplish this classification task, they first extract the opinion terms and the frequent features of the product from the reviews. Popescu and Etzoi [20] also conducted similar research. They first made use of the extraction system called KnowItAll [8] to extract the explicit features of the product. Next the extracted explicit features are utilized to identify the opinion or orientation from the reviews. Both of these two methods apply linguistic techniques and focus on the sentences which are largely grammatical. Our previous work attempted to extract and summarize the product feature and the associated feature values of the hot items from multiple auction Web sites [26]. The idea of this previous work is to extract the product feature and the associated feature values using Hidden Markov models. Next a graph mincut algorithm is applied to identify the hot items in the auction sites by considering the extracted data. All of these existing methods suffer from one major shortcoming in that the extraction task and mining task are conducted without interaction. It is likely the error made in the extraction process accumulates in the mining task.

3 Our Proposed Model

In our framework, we formulate the information extraction and feature mining problem as a graph labeling problem using Conditional Random Fields (CRF). CRF is a discriminative framework based on undirected graphical model [15]. Unlike Naive Bayesian model or other generative models which assume the features are independent, CRF allows the use of a number of overlapping or dependent features. The second advantage is that undirected graphs can model the inter-dependence

between entities, without the need of knowing the actual causality of them. Moreover, many literatures also demonstrate that discriminative models have a promising performance in practice.

A Web page is composed of several informative blocks. An informative block contains text fragments with similar purpose. For example, some informative blocks are about the product features, while some blocks are about advertisement. Therefore, a single Web page actually consists of a set of text fragments and each text fragment can be regarded as a sequence of tokens. In information extraction, the tokens are labeled with different tags. For example, we can design tag labels such as “product feature”, “product feature value”, “normal text”, etc for the digital camera domain. The problem is reduced to labeling the appropriate tag for each token. An undirected graph can be *automatically* constructed representing the conditional dependence among the observation, the tag labels, and various information such as the importance of the tokens. Next, the labeling can be accomplished by conducting inference on the graph structure. To fully automate the information extraction and feature mining task, we also develop an adaptive learning method which can automatically learn the parameters in an unsupervised manner. Human effort in preparing training examples can then be significantly reduced.

We describe below in detail our proposed model in the context of important product feature mining problem. This model can be easily refined to other applications.

3.1 Undirected Graphical Model In CRF, each node in the graph represents a variable and each edge represents the inter-dependence between the connected variables. Suppose we collect a set of Web pages about a product from different Web sites and we wish to discover important product features. Figure 3 shows a simplified CRF model automatically constructed for the important feature mining application. The size of the graph is much larger when dealing with real data. There are two kinds of nodes. The shaded nodes represent observable variables while the unshaded nodes represent unobservable variables. Suppose we have a collection of Web pages \mathfrak{P} . As mentioned above, a Web page, $M \in \mathfrak{P}$, can be regarded as a set of text fragments denoted by \mathfrak{S}^M and each text fragment is considered as a sequence of tokens. For a particular sequence $A \in \mathfrak{S}^M$, each token is actually composed of two kinds of information. The first kind of information is the observation of the tokens such as the word or the capitalization. This information can be observed and is represented by the observable

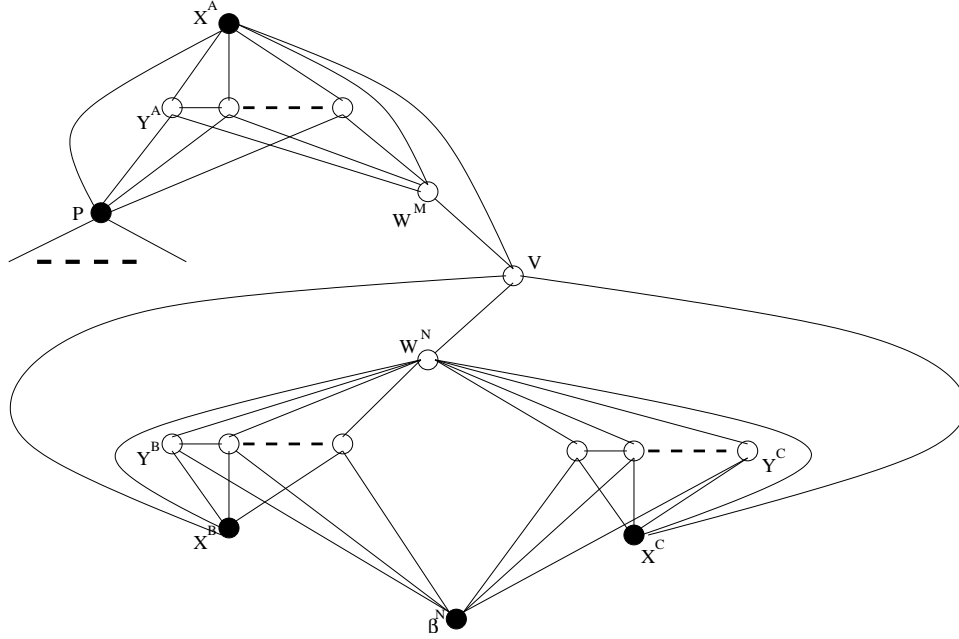


Figure 3: Our proposed conditional random fields model for product feature extraction and important feature mining from multiple Web pages. (Note that P is connected to all the X and Y and the edges between P and each X and Y are not fully shown to avoid clumping for clear presentation.)

variable X^A . The second kind of information is the labeling information of the token. In product feature extraction, each token is labeled with one of the three tags - *product feature*, *product feature value*, and *normal text*. This information is hidden and is represented by the unobservable variable Y^A . Notice that X^A and Y^A actually represent a sequence of variables X_i^A and Y_i^A respectively where $0 < i < L$ and L denotes the number of the tokens in the sequence A . The prior knowledge of the domain is represented by P in our graphical model. Each Y_i^A is connected to Y_{i-1}^A , Y_{i+1}^A , X^A , and P as shown in Figure 3 since the tag label of each token is inter-dependent with the tag labels of the neighbouring tokens, the observation of the sequence, and the prior knowledge. An observable variable denoted by β^M in Figure 3 represents an informative block of a Web page M . It is connected to the sequences located in the same informative block because the likelihood that the text fragments located in the same informative block have similar importance is high. In Figure 3, the sequence $B, C \in \mathfrak{S}^N$ are in the same informative block of the page $N \in \mathfrak{P}$ and $N \neq M$. To automatically determine informative blocks of a Web page, we employ our previous information-theoretic approach [24]. Another variable denoted by W^M refers to the mentioned feature tuples in the Web page M . A feature tuple is defined as the product feature and the associated value. For

instance in Figure 1, one feature tuple contains product feature “camera type” and the associated value “SLR”. Another tuple contains the product feature “resolution” and the associated value “6.1-megapixel”. Obviously, the mentioned feature tuple are inter-dependent with the observation of the sequences and the tag labels of the tokens. The important feature tuples are represented by the variable V . It is inter-dependent with the mentioned feature tuples, as well as the observation of the Web pages because normally the important feature tuples are mentioned by many different Web sites and have some special layout format such as in bolded text.

Once the undirected graph is constructed, the conditional probability of a particular configuration of the hidden variables, given the values of all the observed variables can be written as follows:

$$P(y|x) = \frac{1}{Z} \prod_{C(x,y) \in \mathfrak{C}(x,y)} \Phi(C(x,y)) \quad (3.1)$$

where x and y are the set of observable variables and the set of unobservable variables respectively, $\mathfrak{C}(x,y)$ refers to the set of cliques of the graph. A clique is defined as the maximum complete subgraph. $\Phi(C(x,y))$ refers to the clique potential for $C(x,y)$. Z is called the partition function defined as:

$$Z = \sum_y \prod_{C(x,y) \in \mathfrak{C}(x,y)} \Phi(C(x,y)) \quad (3.2)$$

We define the clique potential as a linear exponential function as follows:

$$\Phi(C(x, y)) = \exp \sum_i \gamma_i f_i(x, y) \quad (3.3)$$

where $f_i(x, y)$ and γ_i are the i -th binary feature and the associated weight respectively. For example, $f_i(x, y)$ equals to one if the underlying token is “resolution” and the class label is product feature and equals to zero otherwise in the digital camera domain. Hence, Equation 3.1 can be written as follows:

$$P(y|x) = \frac{1}{Z} \exp \sum_i \gamma_i f_i(x, y) \quad (3.4)$$

Given the set of γ_i , one can find the optimal labeling of the unobserved variables of the graph via conducting inference. The graph typically consists of a large number of combination for the labels of all the unobservable variables. Hence, direct computation of the probability of a particular labeling of the unobservable variables is infeasible. The inference can be computed by the message passing algorithm, also known as the sum-product algorithm, by transforming the graph using junction tree or factor graph [12]. By finding the configuration of the hidden variables achieving the highest conditional probability stated in Equation 3.1, the desired important product features and the product features values can then be discovered from these Web pages.

3.2 Adaptive Parameter Learning Learning in CRF refers to estimating the value of the weights γ_i associated with each f_i in Equation 3.4. Suppose we have a set of training examples denoted by Tra for which the actual labels of the variables are known. We define the log likelihood function as follows:

$$\mathcal{L}(\gamma_i) = \sum_{j=1}^{j < |Tra|} \left\{ \sum_i \gamma_i f_i(x^{(j)}, y^{(j)}) - \log(Z) \right\} \quad (3.5)$$

where $|Tra|$ and $(x^{(j)}, y^{(j)})$ denotes the number of training examples and the j -th training example respectively. Maximum likelihood approach aims at finding the set of γ_i which maximize Equation 3.5. It can be shown that Equation 3.5 is convex and achieves maximum when the following condition holds:

$$\begin{aligned} \frac{\nabla \mathcal{L}(\gamma_i)}{\nabla \gamma_i} &= \sum_{j=1}^{j < |Tra|} f_i(x^{(j)}, y^{(j)}) \\ &\quad - \sum_{j=1}^{j < |Tra|} \sum_{y'} f_i(x^{(j)}, y^{(j)}) P(y'|x^{(j)}) \\ &= 0 \end{aligned} \quad (3.6)$$

Therefore, one can obtain the set of γ_i achieving the maximum of Equation 3.5 by using iterative methods such as conjugate gradient methods or voted perceptron algorithm. In particular, Figure 4 shows the

Original voted perceptron algorithm for learning CRF

Input: Training examples: Tra

Number of iteration: K

Learning rate: ρ

Initial parameter set: γ_i^0

Output: The final parameter set: γ_i^K

Algorithm:

1. for $k = 0 \dots K - 1$
 2. for $j = 1 \dots |Tra|$
 3. $\hat{y}^{(j),k} = \arg \max_{y'} P(y'|x^{(j)}; \gamma_i^k)$
 4. $\gamma_i^{k+1} \leftarrow \gamma_i^k + \rho \left\{ f_i(x^{(j)}, y^{(j)}) - f_i(x^{(j)}, \hat{y}^{(j),k}) \right\}$
 5. $k \leftarrow k + 1$
 6. end for
 7. end for
 8. return γ_i^K
-

Figure 4: The outline of the supervised voted perceptron learning algorithm for CRF.

outline of voted perceptron for learning the parameters. In essence, the voted perceptron algorithm estimates the weight by iteratively minimizing the following expression:

$$\left| \sum_{j=1}^{j < Tra} f_i(x^{(j)}, y^{(j)}) - \sum_{j=1}^{j < Tra} f_i(x^{(j)}, \hat{y}^{(j)}) \right| \quad (3.7)$$

where $\hat{y}^{(j)}$ is the predicted labeling using the current weighting.

However, recall that our approach is an unsupervised learning. The actual labels of the unobservable variables are not known. We cannot apply the above existing methods for learning the weight. To tackle this problem, we develop an expectation-maximization (EM) based voted perceptron algorithm as shown in Figure 5. In the E-step of our algorithm, we estimate the probability of the labeling of the unobservable variables. In the M-step, we employ the voted perceptron algorithm augmented with the following weight updating function:

$$\gamma_i^{k+1} \leftarrow \gamma_i^k + \rho \left\{ \sum_{y'} f_i(x^{(j)}, y') P(y'|x^{(j)}; \gamma_i^k) - f_i(x^{(j)}, \hat{y}^{(j),k}) \right\} \quad (3.8)$$

Compared with the algorithm stated in Figure 4, our EM based voted perceptron algorithm estimates the weight by iteratively diminishing the following expres-

EM based voted perceptron algorithm for learning CRF

Input: Training examples: Tra

Number of iteration: K

Learning rate: ρ

Initial parameter set: γ_i^0

Output: The final parameter set: γ_i^K

Algorithm:

1. $\gamma_i^* \leftarrow \gamma_i^0$

2. until convergence

E-step:

3. for $j = 1 \dots |Tra|$

4. $P(y'|x^{(j)}) = \frac{1}{Z} \exp \sum_i \gamma_i^* f_i(x^{(j)}, y')$

5. end for

M-step:

6. for $k = 0 \dots K - 1$

7. for $j = 1 \dots |Tra|$

8. $\hat{y}^{(j),k} = \arg \max_{y'} P(y'|x^{(j)}; \gamma_i^k)$

9. $\gamma_i^{k+1} \leftarrow \gamma_i^k +$

$\rho \left\{ \sum_{y'} f_i(x^{(j)}, y') P(y'|x^{(j)}) - f_i(x^{(j)}, \hat{y}^{(j),k}) \right\}$

10. $k \leftarrow k + 1$

11. end for

12. end for

13. $\gamma_i^* \leftarrow \gamma_i^K$

14. end until

15. return γ_i^*

Figure 5: The outline of our unsupervised EM based voted perceptron learning algorithm for CRF. Note that the actual labels of the unobservable variables in Tra are not known.

sion:

$$\left| \sum_{j=1}^{j < Tra} \sum_{y'} f_i(x^{(j)}, y^{(j)}) P(y'|x^{(j)}; \gamma_i^*) - \sum_{j=1}^{j < Tra} f_i(x^{(j)}, \hat{y}^{(j)}) \right| \quad (3.9)$$

The first term of Equation 3.9 (i.e., $\sum_{j=1}^{j < Tra} \sum_{y'} f_i(x^{(j)}, y^{(j)}) P(y'|x^{(j)}; \gamma_i^*)$) is the expectation value of $f_i(x^{(j)}, y')$ and it approaches to the first term of Equation 3.7 (i.e., $\sum_{j=1}^{j < Tra} f_i(x^{(j)}, y^{(j)})$) when the data set is sufficiently large. However, this formulation is no longer convex and can result in a local optimal solution depending on the initial set of parameters. We tackle this program by considering the prior knowledge when choosing the initial parameters.

3.3 Incorporating Prior Knowledge Since our EM based voted perceptron algorithm may result in a

local optimal instead of global solution, choosing the initial set of weights becomes essential. Prior knowledge is considered when choosing the initial weights. There are two types of prior knowledge as described above. The first type is the attribute lexicons. For example, users can easily provide a small set of common product attributes such as “resolution”, “pixel” for the digital camera domain. The second type of prior knowledge is related to the layout format of the Web pages. Normally, the important product features are displayed in some easily perceivable layout format such as in bolded text or at the top position of the Web page. We describe below how to incorporate the prior knowledge into our model.

Recall that CRF is characterized by the set of binary features and the associated weights in Equation 3.4. Prior knowledge can then be easily incorporated by choosing the initial value of the weights before invoking our EM based voted perceptron algorithm. For instance, we can have a larger initial value for γ_i if $f_i(x, y)$ denotes the binary feature that equals to one if the token is “resolution” and the label is “product feature”. We introduce a set of such feature functions about the relationship between the tokens, the lexicon, and the tag labels. Suppose we have a lexicon in which each entry is denoted by e . We can find the *edit distance* normalized by the largest number of characters among e and tok , namely $dist(e, tok)$, between each entry e and the underlying token denoted by tok . Then for each lexicon entry, we define one binary feature function that equals to one if $dist(e, tok)$ is less than a pre-defined threshold θ and the tag label is that entity, and equals to zero otherwise. For example, one entry in the lexicon about digital camera is “resolution”. Then we have one binary feature function that equals to one if $dist(\text{“resolution”}, tok)$ is less than $\theta = 0.3$ and the tag label is “product feature”. The initial weights for these associated feature functions are then set to a higher value. In our experiment, θ and the initial weights are set to 0.3 and 1.0 respectively.

Another type of prior knowledge is related to the layout format of the Web pages. Similarly, we define a set of feature functions about the relationship between the layout format, the mentioned feature tuple, and the important feature tuple. For example, we define a function that equals to one if the tokens are considered as mentioned feature tuple and important feature tuple, as well as located in the upper part of the Web page, and equals to zero otherwise. The initial weights for such kind of feature functions are set to a higher value. In our experiment, the initial weights are set to 1.0.

Product Feature	Product Feature Value
Resolution	6.1 megapixels
Included lens	DX Zoom-Nikkor lens
Camera type	SLR
LCD	1.8"
Electronic flash	i-TTL Speedlight
Image size	3008 x 2000 pixels
Autofocus	5 sensor autofocus
Shooting modes	Portrait, Landscape, Close-Up, Sports, Night Portrait, Night Landscape & Auto

Table 1: The important product features and feature values extracted from the Web pages for camera Nikon D70 depicted in Figures 1 and 2.

Site Label	Web site (URL)
S1	Nice Electronics (www.niceelectronics.com)
S2	OneCall (www.onecall.com)
S3	PCNation (www.pcnation.com)
S4	ZipZoomFly (www.zipzoomfly.com)
S5	Abe's Of Maine (www.abesofmaine.com)
S6	Beach Camera (www.beachcamera.com)
S7	Butterfly Photo (www.butterflyphoto.com)
S8	Buy.com (www.buy.com)
S9	Buydig.com (www.buydig.com)
S10	Newegg.com (www.newegg.com)
S11	2shopper.com (www.2shopper.com)
S12	CompSource (www.c-source.com)
S13	ComputerHQ.com (www.computerhq.com)
S14	Digital Foto Discount Club (www.digitalfotoclub.com)
S15	FuturePowerPC.com (www.futurepowerpc.com)
S16	IBuyDigital.com (www.ibuydigital.com)
S17	Mwave.com (www.mwave.com)
S18	PC Universe (www.pcuniverse.com)
S19	SecureMart.com (www.securemart.com)
S20	TigerDirect (www.tigerdirect.com)

Table 2: The Web sites from where the Web pages were collected for experiments.

4 A Case Study

Consider the two Web pages shown in Figures 1 and 2. These two Web pages are related to the digital camera called “Nikon D70”, but collected from two different retailer Web sites. Each of the Web pages lists several features about the digital camera. For example, the page shown in Figure 2 consists of more than twenty product features such as “6.1 megapixels” for resolution, “3008 X 2000 pixels” for image size, “proprietary battery” for battery type, etc. To identify the important feature of this digital camera, one can browse these two Web pages manually and analyze the features listed. However, it is time-consuming, tedious, and requires high-level of expertise to analyze all the features listed and identify the important ones. To automate this task, we employ our method to handle these two pages. Table 1 shows some of the extracted important features from each of the Web page. These features can effectively describe some special or outstanding features of the product. On the other hand, some features are listed in the Web pages, but are not regarded as important features. For instance, Figure 2 contains the weight and the dimensions of this camera. However, they are not regarded as important features because they are only listed in the detailed specification. Moreover, the weight and dimensions are not the outstanding characteristics of this camera compared with others. Compared with the manually extracted important features tuple, our approach achieves a satisfactory result. The recall and precision are 87.5% and 83.2% respectively¹. This illustrates that our approach can discover useful knowledge for users.

Product label	Product name	Web site
D1	CANON 20D	S1,S3,S4,S5,S6,S7,S8,S10
D2	CANON SD400	S1,S3,S4,S5,S6,S7,S8,S10
D3	CASIO EX-Z750	S1,S5,S7,S8,S9,S10
D4	FUJI F10	S1,S4,S5,S6,S7,S9
D5	KODAK Z740	S1,S3,S5,S6,S7,S8,S9
D6	NIKON D50	S1,S2,S6,S7,S9
D7	NIKON S1	S2,S3, S4,S7,S8,S9S10
D8	PANASONIC DMC-FZ30K	S1,S2,S5,S6,S7,S9
D9	SONY DSC-H1	S1,S2,S5,S7,S9,S10
D10	SONY DSC-T33	S1,S3,S4,S5,S7,S9
M1	Apple iPod 20GB 4th	S10,S13,S15,S18,S19
M2	Archos Gmini XS 200	S8,S10,S12,S17,S18
M3	Creative MOMAD Jukebox	S12,S13,S14,S15,S17,S18,S19
M4	Creative MuVo TX FM	S12,S14,S15,S17,S18
M5	Creative Zen Micro	S12,S13,S14,S15,S17,S18,S19
M6	Lexar LDP-600	S12,S13,S14,S17,S18,S19,S20
M7	Olympus MR-100	S11,S12,S14,S16,S18,S19
M8	Samsung YP-T7Z	S8,S10,S11,S14,S15
M9	SanDisk SDMX1-512-A18	S11,S12,S13,S14,S17,S18,S20
M10	SanDisk Sansa	S12,S13,S17,S18,S19,S20

Table 3: The products used in the experiment and the Web sites from where the Web pages were collected.

5 Experimental Results

We have conducted extensive experiments to demonstrate the effectiveness of our framework. As mentioned before, we have applied our model to the application namely important product feature mining from retailer Web sites. It aims at extracting the product features and the associated feature values as well as discovering the important feature tuples simultaneously. We define an important feature tuple as the feature tuple which is either located in the foremost viewable position, in bolded text, italic, or in color different from the normal majority texts in the Web page. Foremost viewable position refers to the position of the Web page which can be seen by the users without scrolling. We conducted experiments for two different domains, namely, digital camera domain, and MP3 player domain. In each domain, we randomly chose ten different products. For a particular product, we collected Web pages describing the product from different source Web sites. Table 2 depicts the Web sites from where the Web pages used in the experiments were collected. The first and second columns of the table represent the Web site labels, and the Web sites (URL) respectively. Table 3 shows the products used in the experiment. The third column depicts the Web sites from which the Web pages were collected. Note that the Web pages were not collected from the same set of source Web sites because some products cannot be found in certain retailer sites. The product labeled with D1 - D10 were collected for the digital camera domain and the products labeled with M1 - M10 were collected for the MP3 player domain. We manually annotated the product features and product feature values on the Web pages and the annotations are treated as the gold standard for evaluation purpose. Next, we apply our approach to the Web pages of each product to identify the important feature. For example, all the 10 pages about the digital camera “Canon EOS 20D” were considered during extraction and mining using our unified model. As described in Section 3.3, a small attribute lexicon was used for the prior knowledge.

We adopt two metrics, namely, precision and recall to measure the performance. Precision is defined as the number of items correctly extracted divided by the total number of extracted items. Recall is defined as the number of items correctly extracted divided by the total number of actual items. We have also conducted experiments on two existing unsupervised Web mining tech-

¹The definitions of recall and precision are presented in Section 5.

niques, namely, MDR² [16] and ROADRUNNER³ [7] for comparison.

Since MDR and ROADRUNNER do not consider the importance in extraction, the data extracted by MDR and ROADRUNNER are regarded as the important features. Table 4 shows performance of our approach in the digital camera domain. In this tables, the first column shows the Web page label from where the information is extracted. The column labeled with “Our Approach” shows the extraction performance for each product using our approach. The third and fourth columns depict the extraction performance of MDR and ROADRUNNER respectively. Our approach obtains an average recall and precision about 71.8% and 64.7% respectively. MDR and ROADRUNNER fail to effectively discover important features⁴. Table 5 presents the extraction performance in the MP3 player domain. The results are similar to that of the digital camera domain, with both average recall and precision about 61.4% and 59.7% respectively. ROADRUNNER and MDR fail to extract good quality information because these two methods rely on the repeated patterns in the Web pages. However, a Web page from the retailer Web sites normally contains only a single product and is lack of repeated patterns. The experimental results illustrate that our approach can discover the important features from multiple Web pages without providing any training examples. Our work can be very effective for discovering useful information.

6 Conclusions and Future Work

We have developed a framework which can automatically extract information and discover useful knowledge from multiple Web page collaboratively in a unified model. Our approach can reduce the boundary between information extraction and data mining and allow tight interaction of the tasks. Our framework is designed based on the undirected graphical model called conditional random fields. One characteristic of this undirected graph is that it can model the inter-dependence between the neighbouring tokens inside a single Web page, as well as the tokens from different Web pages. Another characteristic is that it allows conducting feature mining across Web pages simultaneously. We have applied our framework in the application namely im-

²The software package of MDR can be obtained in the URL: <http://www.cs.uic.edu/liub/WebDataExtraction/MDR-download.html>.

³The software package of ROADRUNNER can be obtained in the URL: <http://www.dia.uniroma3.it/db/roadRunner/index.html>.

⁴Since ROADRUNNER fails to generate the wrappers in most of the cases, the extraction performance is unsatisfactory.

	Our Approach		MDR		ROADRUNNER	
	R	P	R	P	R	P
D1	71.2	73.3	85.3	13.2	2.1	3.5
D2	71.7	66.7	70.2	15.3	1.5	2.3
D3	66.7	55.4	66.6	13.3	2.4	2.6
D4	78.5	64.3	73.4	15.9	4.5	3.8
D5	63.3	57.7	77.6	16.4	1.5	3.2
D6	76.7	62.2	80.4	12.4	2.4	2.7
D7	68.1	69.6	70.5	19.4	2.8	1.6
D8	78.5	56.0	78.6	12.5	1.9	1.7
D9	76.9	68.3	79.1	15.4	2.3	2.1
D10	66.8	73.3	70.1	13.4	1.7	1.4
Ave	71.8	64.7	75.2	14.7	2.3	2.5

Table 4: The experimental results of our approach, MDR, and ROADRUNNER for identifying the important features for the digital camera domain in the important feature mining application. (R and P refer to recall and precision respectively. Ave. refers to the average extraction performance.)

	Our Approach		MDR		ROADRUNNER	
	R	P	R	P	R	P
M1	65.3	54.2	70.3	15.4	3.4	1.5
M2	51.4	61.2	60.9	17.5	4.5	2.1
M3	55.9	47.7	50.3	12.4	2.1	1.6
M4	61.4	62.0	65.5	17.1	3.8	2.4
M5	61.7	58.4	70.9	13.4	3.7	1.2
M6	57.3	63.4	69.0	13.1	2.3	1.7
M7	60.3	53.7	63.3	12.1	3.4	5.4
M8	70.1	65.3	77.8	14.5	1.9	2.4
M9	63.5	67.7	78.0	13.8	2.8	2.1
M10	66.9	63.4	77.8	54.9	1.8	3.7
Ave	61.4	59.7	68.4	18.4	3.0	2.4

Table 5: The experimental results of our approach, MDR, and ROADRUNNER for identifying the important features for the MP3 player domain in the important feature mining application. (R and P refer to recall and precision respectively. Ave. refers to the average extraction performance.)

portant product feature mining. Extensive experiments have been conducted to demonstrate the effectiveness of our approach.

We intend to extend our framework in several directions. One possible direction is to apply our framework to other applications such as mining the domain ontology. Recently, semantic Web, which is considered as the next generation of Web, becomes an active research area [27]. Ontology is an essential component in semantic Web because it contains the domain knowledge. Normally, this ontology is manually constructed by human experts. Our previous work attempts to refine an existing ontology of a Web site to adapt to other previously unseen sites [25]. We intend to employ our approach to automatically extract the information from multiple Web pages and construct the ontology simultaneously. Another direction is to investigate the usage of an existing ontology of the domain or some constraints on the extracted data, to refine the extraction results.

References

- [1] E. Agichtein and V. Ganti. Mining reference tables for automatic text segmentation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 20–29, 2004.
- [2] J. Ambite, G. Barish, C. Knoblock, M. Muslea, J. Oh, and S. Minton. Getting from here to there: Interactive planning and agent execution for optimizing travel. In *Proceedings of the Fourteenth Innovative Applications of Artificial Intelligence Conference*, pages 862–869, 2002.
- [3] R. Bunescu and R. Mooney. Collective information extraction with relational markov networks. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 439–446, 2004.
- [4] C. Chang and S. C. Lui. IEPAD: information extraction based on pattern discovery. In *Proceedings of the Tenth International Conference on World Wide Web (WWW)*, pages 681–688, 2001.
- [5] F. Ciravegna. $(LP)^2$ an adaptive algorithm for information extraction from web-related texts. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1251–1256, 2001.
- [6] V. Crescenzi and G. Mecca. Automatic information extraction from large websites. *Journal of the ACM*, 51(5):731–779, 2004.
- [7] V. Crescenzi, G. Mecca, and P. Merialdo. ROADRUNNER: Towards automatic data extraction from large web sites. In *Proceedings of the 27th Very Large Databases Conference (VLDB)*, pages 109–118, 2001.
- [8] O. Etzioni, M. Cafarella, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Un-supervised named-entity extraction from the web: An

- experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.
- [9] D. Freitag and A. McCallum. Information extraction with HMM structures learned by stochastic optimization. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI)*, 2000.
- [10] T. Grenager, D. Klein, and C. Manning. Unsupervised learning of field segmentation models for information extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 371–378, 2005.
- [11] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 168–177, 2004.
- [12] F. Kschischang, B. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transaction on Information Theory*, 47(2):498–519, 2001.
- [13] N. Kushmerick and B. Grace. The wrapper induction environment. In *Proceedings of the Workshop on Software Tools for Developing Agents (AAAI)*, pages 131–132, 1998.
- [14] N. Kushmerick and B. Thomas. Adaptive information extraction: Core technologies for information agents. In *Intelligent Information Agents R&D In Europe: An AgentLink Perspective*, pages 79–103, 2002.
- [15] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289, 2001.
- [16] B. Liu, R. Grossman, and Y. Zhai. Mining data records in web pages. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 601–606, 2003.
- [17] A. McCallum and D. Jensen. A note on the unification of information extraction and data mining using conditional-probability, relational models. In *Proceedings of the IJCAI Workshop on Learning Statistical Models from Relational Data*, 2003.
- [18] A. McCallum and B. Wellner. Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proceedings of the IJCAI Workshop on Information Integration on the Web*, 2003.
- [19] I. Muslea, S. Minton, and C. Knoblock. Hierarchical wrapper induction for semistructured information sources. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1-2):93–114, 2001.
- [20] A. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Human Language Technology Conference Conference on Empirical Methods in Natural Language Processing (2005)*, 2005.
- [21] P. Viola and M. Narasimhan. Learning to extract information from semi-structured text using a discriminative context free grammar. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development In Information Retrieval (SIGIR)*, pages 330–337, 2005.
- [22] B. Wellner, A. McCallum, F. Peng, and M. Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 593–601, 2004.
- [23] T. L. Wong and W. Lam. A probabilistic approach for adapting information extraction wrappers and discovering new attributes. In *Proceedings of the 2004 IEEE International Conference on Data Mining (ICDM)*, pages 257–264, 2004.
- [24] T. L. Wong and W. Lam. Text mining from site invariant and dependent features for information extraction knowledge adaptation. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)*, pages 45–56, 2004.
- [25] T. L. Wong and W. Lam. Learning to refine ontology for a new web site using a bayesian approach. In *Proceedings of the 2005 SIAM International Conference on Data Mining (SDM)*, pages 298–309, 2005.
- [26] T. L. Wong and W. Lam. Hot item mining and summarization from multiple auction web sites. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM)*, To appear, 2005.
- [27] World Wide Web Consortium (W3C). Semantic web. In <http://www.w3.org/2001/sw/>, 2001.