

An Unsupervised Framework for Extracting and Normalizing Product Attributes from Multiple Web Sites

ABSTRACT

We have developed an unsupervised framework for simultaneously extracting and normalizing attributes of products from multiple Web pages originated from different sites. Our framework is designed based on a probabilistic graphical model that can model the page-independent content information and the page-dependent layout information of the text fragments in Web pages. One characteristic of our framework is that previously unseen attributes can be discovered from the clue contained in the layout format of the text fragments. Our framework tackles both extraction and normalization tasks by jointly considering the relationship between the content and layout information. Dirichlet process prior is employed leading to another advantage that the number of discovered product attributes is unlimited. An unsupervised inference algorithm based on variational method is presented. The semantics of the normalized attributes can be visualized by examining the term weights in the model. Our framework can be applied to a wide range of Web mining applications such as product matching and retrieval. We have conducted extensive experiments from four different domains consisting of over 300 Web pages from over 150 different Web sites, demonstrating the robustness and effectiveness of our framework.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Web mining, attribute extraction, attribute normalization

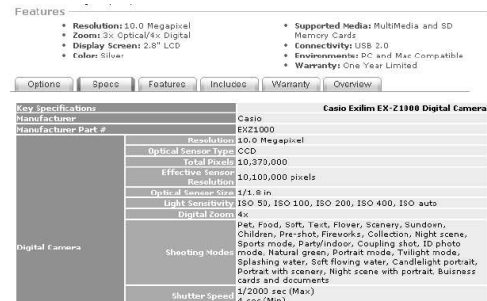
1. INTRODUCTION

The World Wide Web (WWW) contains a huge number of online stores selling million of different kinds of products. While online stores can reduce the geographical barrier and the time constraint for shopping, it becomes problematic for

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WOODSTOCK '97 El Paso, Texas USA

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.



The screenshot shows a web page with a navigation bar at the top containing tabs: Options, Specs, Features, Includes, Warranty, and Overview. The 'Features' tab is selected. Below the navigation bar, there are two columns of bullet points listing features. The left column includes: Resolution: 10.0 Megapixel, Zoom: 3x Optical/4x Digital, Display screen: 2.8" LCD, and Color: Silver. The right column includes: Supported Media: MultiMedia and SD Memory Cards, Connectivity: USB 2.0, Environment: DC and Mac Compatible, and Warranty: One Year Limited. Below the features, there is a section titled 'Key Specifications' for the 'Casio Exilim EX-Z1100 Digital Camera'. This section contains a table with the following specifications: Manufacturer: Casio, Manufacturer Part #: EXZ1000, Resolution: 10.0 Megapixel, Optical Sensor Type: CCD, Total Pixels: 10,375,000, Effective Sensor Resolution: 10,100,000 pixels, Optical Zoom: 4x, Digital Zoom: 4x, Light Sensitivity: ISO 50, ISO 100, ISO 200, ISO 400, ISO auto, Shooting Modes: Portrait, Food, Soft, Text, Flower, Scenery, Sundown, Children, Per-shot, Fireworks, Collection, Night scene, Sports mode, Party/indoor, Coupling shot, ID photo mode, Natural green, Portrait mode, Twilight mode, Splashing water, Soft flowing water, Candlelight portrait, Portrait with zoomer, Night scene with portrait, Business cards and documents, and Shutter Speed: 1/2000 sec (Max) to 4 sec (Min).

Key Specifications	
Manufacturer	Casio
Manufacturer Part #	EXZ1000
Resolution	10.0 Megapixel
Optical Sensor Type	CCD
Total Pixels	10,375,000
Effective Sensor Resolution	10,100,000 pixels
Optical Zoom	4x
Digital Zoom	4x
Light Sensitivity	ISO 50, ISO 100, ISO 200, ISO 400, ISO auto
Shooting Modes	Portrait, Food, Soft, Text, Flower, Scenery, Sundown, Children, Per-shot, Fireworks, Collection, Night scene, Sports mode, Party/indoor, Coupling shot, ID photo mode, Natural green, Portrait mode, Twilight mode, Splashing water, Soft flowing water, Candlelight portrait, Portrait with zoomer, Night scene with portrait, Business cards and documents
Shutter Speed	1/2000 sec (Max) to 4 sec (Min)

Figure 1: A sample of a portion of a Web page showing some product information of a digital camera collected from a Web site. (Web site URL: <http://www.superwarehouse.com>)

a user to retrieve, analysis, and compare products. For example, Figure 1 shows a sample of a portion of a Web page about some product information of a digital camera which consists of several product attributes such as resolution, flash mode, etc. Traditional search engines whose retrieval methods treat every term in a Web document in a uniform fashion often result in ineffective product attribute information extraction and analysis. For example, in the digital camera domain, if we supply the query terms “auto white balance”, existing search engines may just match the terms in Web pages and return products without attribute content “auto white balance” but with attribute content “auto ISO”. In fact, “auto white balance” and “auto ISO” refer to the values of the attribute “white balance” and “light sensitivity”, which should be regarded as two different attributes of a digital camera, and they should be properly differentiated. Manually identifying product attributes is tedious and time consuming, and practically infeasible for the massive amount of Web sites. As a result, it raises the need for automated methods which can identify the attributes of products effectively from Web pages. If the product attribute information is extracted from multiple Web sites, another desirable task is that the product attributes can be automatically normalized and preferably the semantic meaning of normalized attributes can be obtained. This can improve the indexing of product Web pages, and support further intelligent tasks such as attribute search or product matching.

Existing information extraction approaches aim at extracting precise text fragments from documents [11]. In particular wrapper learning techniques have been developed to extract information from semi-structured documents such as Web pages [18]. For example, by collecting training examples, which consists of certain product attributes, from some Web pages in the site shown in Figure 1, one can learn



Figure 2: A sample of a portion of a Web page showing some product information of a digital camera collected from a Web site different from the one depicted in Figure 1. (Web site URL: <http://www.crayeon3.com>.)

a wrapper for automatically extracting information from the remaining pages in the *same* Web site. However, one major limitation of existing wrapper learning methods is that they are supervised method and hence it requires manual effort in preparing training examples for every product attribute. Moreover, the learned wrapper can only be applied to the Web site where the training examples come from. For instance, the learned wrapper for Figure 1 cannot be applied to the Web site shown in Figure 2 because of essentially different layout formats of the two sites. A separate human work is needed to prepare a new set of training examples. As a consequence, existing wrapper construction or learning methods are not scalable if we wish to extract information from numerous different Web sites.

Recently, Zhu et al. have developed a system which can segment Web pages and label the elements of Web pages from different sources [19]. Their method analyzes the layout format of Web pages and employ an integrated approach of Hierarchical Conditional Random Fields (HCRF) and Semi-Conditional Random Fields (Semi-CRF) for segmenting and labeling the text fragments in a single framework. Their approach is template dependent and can be applied to multiple Web sites for information extraction. Though this integrated method can partially solve the problem of wrapper learning, one limitation is that it is a supervised learning method and one has to define the set of possible product attributes in advance and provide training examples for each attribute. In other words, it cannot handle the discovery of previously unseen attributes.

To reduce the human work involved, several unsupervised wrapper learning techniques have been proposed [5] by making use of the layout format of Web pages which are generated by templates. Since the extraction is template dependent, the data extracted from different sites, even in the same domain, may not be synchronized. For example, a field extracted from a particular site may contain both book title and author, whereas in another site, book title and author correspond to two different extracted fields. Chuang et al. proposed an unsupervised wrapper learning technique which can construct wrappers to extract synchronized data from multiple sources [4]. The objective is to identify the optimal segmentation of the text in Web pages. For example, the field containing both book title and author can be automatically segmented into two separate fields or attributes

However, their method requires to train a field model for each field. For example, there are two different field models for book title and author. These field models are required to be trained from manually prepared training examples, or developed by human experts in advance costing substantial human effort. Moreover, it cannot handle previously unseen fields of records. They proposed a heuristic methodology for training the field models for previously unseen fields in an unsupervised manner. The idea is to consider each group of aligned segments created by an unsupervised wrapper as a single field, and train a field model for each group using HMM with a predefined labeling rule. However, such method can only apply to a Web page that contains multiple records. For a Web pages with a single record, such as the ones in Figures 1 and 2, there exists neither group of aligned segments, nor a single group in which the aligned segments refer to different fields.

Another limitation of existing unsupervised wrapper methods is that the extracted fields from different Web sites are not normalized, and hence requiring human work to judge whether two extracted fields refer to the same attribute in a domain. For example, one may not know that the extracted text fragments “fireworks” are “portrait” refer to two different attribute values of the same attribute “shooting mode” in the digital camera domain. Normalization of attribute is defined as clustering attribute values with similar semantic meaning. It is useful for many applications such as storing attribute values of product records into structured database, retrieving and matching of products, etc. Chuang et al. proposed a clustering method to match the extracted data based on the tokens of the data in a separate step [4]. However, since their method mainly considers the tokens, it is not able to normalize the text fragments “fireworks” and “portrait” to the same attribute. Moreover, the clustering algorithm requires to fix the number of clusters in advance. In practice, the number of attributes in a domain is unknown, and new features are found in a domain from time to time resulting in an unlimited number of attributes.

The requirement of training examples, the incapability of discovering unseen attributes, and the lack of normalization of extracted attributes with similar semantic meaning are the problems of existing approaches. In this paper, we aim at addressing these problems by developing an unsupervised learning framework for jointly extracting and normalizing product attributes from multiple Web sites.

1.1 A Motivating Example

Consider the Web pages shown in Figures 1 and 2. These two Web pages are collected from two different Web sites in the digital camera domain describing two different digital cameras. Naturally, they have different layout formats because they come from different Web sites. We define attribute and attribute value as a field of a product and a value for a particular field respectively. For example one attribute of a digital camera is “sensor resolution” and the attribute values are “effective sensor resolution 10,100,000 pixels” and “sensor resolution 10 megapixels” for the products in Figures 1 and 2 respectively. To extract the product attribute values from these two pages, one can make use of two wrappers, which must be previously learned for each individual Web site, to accomplish the task. However, as described before, human work is needed to prepare training examples for wrapper learning and the attributes to be extracted are required to be defined.

Very often, users may have some prior knowledge about the content of *some* attributes of interest in the domain. For example, users may know that some terms such as “megapixel” and “ISO” are frequently used to describe a digital camera. Such prior knowledge can be easily collected, for example, by scanning one Web page about digital cameras and collecting a few terms in a list. We can utilize the prior knowledge and infer from the content of the text fragments in Web pages that the text fragment “sensor resolution 10 megapixels” in Figures 2 likely refers to an attribute value. However, there may be some previously unseen attributes. For example, from the layout format of the Web page in Figure 2, it can be inferred that the text fragment “white balance auto, daylight, cloudy, tungsten, fluorescent, fluorescent H, custom” should be an attribute value because the layout format of this text fragment share certain similarity to the extracted text fragment “sensor resolution 10 megapixels”. It likely corresponds to an attribute value of a previously unseen attribute “white balance”. Similarly, more attribute values, which correspond to some previously unseen attributes such as “shutter speed”, can be discovered from both Figures 1 and 2. This shows that there is mutual influence between the content and layout format of product attributes in Web pages. This provides useful clues for extracting attribute values of previously unseen attributes.

The above scenario demonstrates the possibility of making use of the layout format of text fragments for extraction. The next issue is the requirement of human effort to interpret the semantic meanings of the attribute values of the previously unseen attributes. For example, suppose there is an extracted text fragment “fluorescent” in other Web pages. The text fragment “white balance auto, daylight, cloudy, tungsten, fluorescent, fluorescent H, custom” is extracted from the Web page shown in Figure 2. Suppose that these two text fragments are automatically clustered to the same group representing an attribute. One can easily observe that they refer to the same attribute corresponding to “white balance”. This allows better understanding and interpretation of the semantic meaning of the normalized attribute because of some indicative terms such as “white balance” appeared in the majority of the text fragments in the same group.

After product attributes are extracted and normalized, a product can be effectively represented. It is useful for indexing the terms and conducting other intelligent tasks such as product matching and comparison.

1.2 Our Contributions

We have developed an unsupervised learning framework for jointly extracting and normalizing product attributes from multiple Web sites. For example, the text fragments “fireworks” are “portrait” are samples of extracted and normalized text fragments in the digital camera domain using our method. These two fragments do not have words in common, but actually they refer to the product attribute “shooting mode” in the digital camera domain. Unlike existing methods which conduct the extraction and normalization tasks in separate steps unavoidably leading to the accumulation of errors in these two steps, we propose a single framework which can conduct extraction and normalization tasks simultaneously resulting in a solution optimizing both tasks. We also demonstrate in our mathematical formulation, that considering both the content information and the layout information can resolve the conflict between the two

tasks.

Our framework considers the *page-independent* content information and the *page-dependent* layout information in a single framework. As illustrated in the above motivating example, the mutual influence between the content and the layout format of text fragments provides useful clues for both of the tasks of attribute extraction and normalization. We design a probabilistic graphical model to model the relationship between the content and layout information for solving the extraction and normalization tasks simultaneously. We employ Dirichlet process prior leading to another characteristic that the number of attributes to be discovered need not to be fixed and can be unlimited. This can handle product attributes not known in advance and new attributes can be discovered.

The semantic meaning of the extracted and normalized attributes can be visualized by a set of weighted terms in the model. This can significantly help users understand and interpret the attributes. Our framework can be applied to applications such as improving product searching based on attributes and Web online product matching. We have conducted extensive experiments from four different domains consisting of over 300 Web pages from over 150 Web sites. The experimental results show that our framework is robust and effective.

2. PROBLEM DEFINITION AND FORMAL MODEL

2.1 Problem Definition

In a product domain \mathcal{D} , we have a set of *reference* attributes, denoted by \mathcal{A} to describe the products. Let a_i be the i -th attribute in \mathcal{A} . For example, in the digital camera domain, reference attributes of digital camera may include “resolution”, “white balance”, “light sensitivity”, etc. There exists a special element denoted as \bar{a} representing “not-an-attribute”. Since the number of attributes is unknown and hence the size of \mathcal{A} denoted by $|\mathcal{A}|$ is between 0 and ∞ . Each product r in \mathcal{D} , is then characterized by the attribute values of the reference attributes. Let $v_i(r)$ be the attribute value of the reference attribute a_i for product r . For instance, the attribute value of the reference attribute “resolution” shown in Figure 1 is ‘10.0 Megapixel’.

Given a collection of product Web pages \mathcal{C} collected from a set of Web sites \mathcal{S} . Let $c_i(s)$ be i -th page collected from the site s . Each page contains a single product p . Within the Web page $c_i(s)$, we can collect a set of text fragments $\mathbf{X}(c_i(s))$. For example, “resolution 10,100,000 pixels” and “optical sensor type CCD” are samples of text fragments collected from the page shown in Figure 1. Let $x_j(c_i(s))$ be the j -th text fragment in the Web page $c_i(s)$. Essentially, each x in $\mathbf{X}(c_i(s))$ can be represented by a four-field tuple (C, L, T, A) . C refers to the *content information* of the text fragment such as the tokens contained in “optical sensor type CCD”. L refers to the *layout information* of the text fragment. For example, the text fragment “Feature” is grey and in larger font size. T , defined as the *target information*, is a binary variable which is equal to 1 if the underlying text fragment is an attribute value, and 0 otherwise. For example, the values of T for the text fragments “Feature” and “optical sensor type CCD” are 0 and 1 respectively. A defined as the *attribute information*, refers to the reference attribute that the underlying text fragment belongs to. It is a realization of \mathcal{A} and hence it must be equal to one of

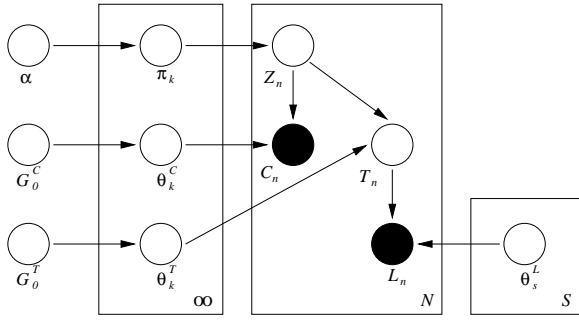


Figure 3: The graphical model for the generation of text fragments in Web pages.

the elements in \mathcal{A} . For example, the values of A for the text fragments “resolution 10,100,000 pixels” and “white balance auto, daylight, cloudy, tungsten, fluorescent, fluorescent H, customoptical” should be equal to the reference attributes “resolution” and “optical sensor” included in \mathcal{A} respectively.

In practice, the content information C and the layout information L of a text fragment can be observed from Web pages. However, the target information T and the attribute information A cannot be observed. As a result, given the observation of C and L , product attribute extraction can be formulated as the prediction for the value of T for each text fragment in Web pages aiming at discovering all text fragments corresponding to certain attribute values. Formally, for each text fragment, we aim at finding $T = t^*$ such that $t^* = \underset{t}{\operatorname{argmax}} \{P(T = t|C, L)\}$. Attribute normal-

ization can be defined as the prediction of the value of A for each text fragment, so that one can understand the reference attribute to which the underlying text fragment refers. Formally, for each text fragment, we aim at finding $A = a^*$ such that $a^* = \underset{a}{\operatorname{argmax}} \{P(A = a|C, L)\}$. When $T = 1$, $P(A = a|C, L) > 0$ for some $a \in \mathcal{A} \setminus \{\bar{a}\}$ and $P(A = \bar{a}|C, L) = 0$. When $T = 0$, $P(A = \bar{a}|C, L) = 1$. Obviously, $P(T|C, L)$ and $P(A|C, L)$ are dependent since $P(A|T = 0, C, L) \neq P(A|C, L)$. As a result, conducting product attribute extraction and normalization separately may lead to conflict solutions degrading the performance of both tasks. In our framework, we aim at predicting the values of T and A such that the joint probability $P(T, A|C, L)$ can be maximized leading to a solution optimizing both tasks.

2.2 Our Model

Our model can be regarded as an extension of Dirichlet mixture model. Each mixture component, which refers to a reference attribute in our framework, consists of its own distribution about text fragments. Dirichlet process prior is employed so that our framework can handle unlimited number of reference attributes. Figure 3 shows the plate diagram representation of our model. Shaded nodes and unshaded nodes represent the observable and unobservable variables respectively. The edges represent the dependence between variables and the plates represent the repetition of variables. We adopt the stick breaking construction representation of Dirichlet process in our presentation [15].

Suppose we have a collection of N different text fragments collected from S different Web pages. Each generation of a text fragment is modeled as an independent and identical event. The n -th text fragment x_n consists of an unobservable variable Z_n depending on the variables

$\pi = \{\pi_1, \pi_2, \dots\}$. Z_n represents the index of the mixture component from which that the underlying text fragment is generated. Essentially, we use Z_n to replace A_n for clarity and $A_n = a_{z_n}$ where $a_i \in \mathcal{A}$. Next, the content information of the text fragment, denoted as C_n , is generated according to $P^C(C_n|\theta_{z_n}^C)$, where $P^C(\cdot|\theta_k^C)$ is the probability distribution about the content C given the variable θ_k^C ; and k refers to the k -th mixture component. On the other hand, the target information T_n is generated by $P^T(T_n|\theta_{z_n}^T)$, where $P^T(\cdot|\theta_k^T)$ is the probability distribution about the target information T given the variable θ_k^T . Since the layout format of the text fragments in a Web page is page-dependent, we have a set of layout distributions, namely, θ_s^L , for generating the layout format of the text fragments in page s . As shown in the running example in Section 1.1, there is mutual influence between the layout information and the target information of a text fragment. T_n together with θ_s^L will generate the layout information L_n of the n -th text fragment according to $P^L(L_n|T_n, \theta_{s(x_n)}^L)$, where $P^L(\cdot|T_n, \theta_s^L)$ is the probability distribution about the layout information L given the variables T_n and θ_s^L ; and $s(x_n)$ denotes the Web page from which x_n is collected.

Unlike ordinary Dirichlet mixture models in literature, in which they consider only one distribution in each mixture component, our framework consists of two different distributions characterized by θ_k^C and θ_k^T for the k -th component. θ_k^C and θ_k^T are basically generated from the base distributions G_0^C and G_0^T respectively in the Dirichlet process. G_0^C and G_0^T act as the prior distributions of the content information and the target information respectively. For example, suppose we model the content of the text fragments by a mixture model of tokens, G_0^C can be a Dirichlet distribution which is the conjugate prior of a mixture model, $P^C(\cdot|\theta_k^C)$ a multinomial distribution, and θ_k^C is the set of parameters of multinomial distribution in component k . Similarly, Since T is a binary variable, it can be modeled as a Bernoulli trial. Therefore, $P^T(\cdot|\theta_k^T)$ can be a binomial distribution with parameter θ_k^T and G_0^T can be a Beta distribution, which is the conjugate prior of a binomial distribution.

Recall that the Dirichlet process is represented by the stick breaking construction in the graphical model depicted in Figure 3. In the stick breaking construction, we have a one-unit length stick and we break a π_k portion from the remaining portion of the stick according to $\mathbf{Beta}(1, \alpha)$ in the k -th break, where $\mathbf{Beta}(\alpha_1, \alpha_2)$ is the Beta distribution, with parameters α_1 and α_2 . The process repeats for infinite times and hence the k -th piece of the broken sticks can represent the proportion of k -th component in the mixture. Therefore, Dirichlet process prior can support an infinite number of mixture components, which refer to the product attributes in our framework. Z_n is then drawn from the distribution π . In summary, the generation process can be described as follows:

$$\begin{aligned} \pi_k | \alpha &\sim \mathbf{Beta}(1, \alpha) & \pi_k &= \tilde{\pi}_k \prod_{i=1}^{k-1} (1 - \tilde{\pi}_i) \\ Z_n | \pi &\sim \pi & \theta_k^T | G_0^T &\sim G_0^T \\ \theta_k^C | G_0^C &\sim G_0^C & C_n | \theta_k^C &\sim P^C(C_n | \theta_{Z_n}^C) \\ T_n | \theta_k^T &\sim P^T(T_n | \theta_{Z_n}^T) & L_n | T_n, \theta_{s(x_n)}^L &\sim P^L(L_n | T_n, \theta_{s(x_n)}^L) \end{aligned}$$

The joint probability for generating a particular text fragment x_n given the parameters α , G_0^C , G_0^T , and θ_s^L can then be expressed as follows:

$$\begin{aligned} &P(C_n, L_n, T_n, Z_n, \pi_1, \pi_2, \dots, \theta_1^C, \theta_2^C, \dots, \theta_1^T, \theta_2^T, \dots | \alpha, G_0^C, G_0^T, \theta_s^L) \\ &= \prod_{i=1}^{\infty} \{P^L(L_n | T_n, \theta_{s(x_n)}^L) [P^C(C_n | Z_n, \theta_{Z_n}^C) P^T(T_n | Z_n, \theta_{Z_n}^T)]^{\chi_{\{Z_n=i\}}}\} \\ &\quad P(Z_n = i | \pi_1, \pi_2, \dots) P(\theta_i^C | G_0^C) P(\theta_i^T | G_0^T) \prod_{i=1}^{\infty} P(\pi_i | \alpha, \pi_1, \dots, \pi_{i-1}) \end{aligned} \quad (1)$$

where $\chi_{\{Z_n=i\}} = 1$ if $Z_n = i$ and 0 otherwise. For simplic-

ity, we let \mathbf{O} , \mathbf{U} , and φ be the set of observable variables, which include all C_n and L_n , the set of unobservable variables, which include all T_n , Z_n , θ_k^C , θ_k^T , and π_k , and the set of model parameters, which include α , \mathbf{G}_0^C , \mathbf{G}_0^T , θ_s^L respectively. Given a set of N text fragment \mathbf{X} and the parameters φ , the inference problem is then defined as follows:

$$\begin{aligned} u^* &= \operatorname{argmax}_u \{P(\mathbf{U} = u | \mathbf{O}, \varphi)\} \\ &= \operatorname{argmax}_u \{\log P(\mathbf{U} = u | \mathbf{O}, \varphi)\} \end{aligned} \quad (2)$$

Since the computation of $\log P(\mathbf{U} | \mathbf{O}, \varphi) = \log \int P(\mathbf{U}, \mathbf{O} | \varphi) d\mathbf{O}$ involves the marginalization of $P(\mathbf{U}, \mathbf{O} | \varphi)$, that is defined in Equation 1, over the unobservable variables, exactly solving Equation 2 is intractable. As a result, approximation methods such as Markov Chain Monte Carlo (MCMC) algorithm are required. In this paper, we develop a variational method to tackle this problem.

3. UNSUPERVISED INFERENCE USING VARIATIONAL METHOD

3.1 Variational Method

Recall that the objective of the inference is to compute $P(\mathbf{U} | \mathbf{O}, \varphi)$, however, it is intractable. The main idea of our method is to design a tractable distribution $Q(\mathbf{U} | \nu)$, which is called the variational distribution of \mathbf{U} characterized by a set of variational parameters denoted as ν . The designed $Q(\mathbf{U} | \nu)$ should be as closer to $P(\mathbf{U} | \mathbf{O}, \varphi)$ as possible. The distance between any two probability distributions P and Q can be measured by Kullback-Leibler(KL) divergence defined as $D(Q || P) = \sum_{x \in X} Q(x) \frac{\log Q(x)}{\log P(x)}$ where X refers to all possible events. Therefore, our method aims at minimizing the following KL-divergence by altering the set of variational parameters:

$$\begin{aligned} D(Q(\mathbf{U} | \nu) || P(\mathbf{U} | \mathbf{O}, \varphi)) &= E_Q[\log Q(\mathbf{U} | \nu)] - E_Q[\log P(\mathbf{U} | \mathbf{O}, \varphi)] \\ &= E_Q[\log Q(\mathbf{U} | \nu)] - E_Q[\log P(\mathbf{U}, \mathbf{O} | \varphi)] + \log P(\mathbf{O} | \varphi) \end{aligned} \quad (3)$$

Since $D(Q || P) \geq 0$, we have:

$$\log P(\mathbf{O} | \varphi) \geq E_Q[\log P(\mathbf{U}, \mathbf{O} | \varphi)] - E_Q[\log Q(\mathbf{U} | \nu)] \quad (4)$$

The left-hand-side (LHS) is the log likelihood of the observation of all text fragments \mathbf{X} given the model parameters, the right-hand-side (RHS) is the lower bound of the likelihood function. The minimization of $D(Q(\mathbf{U} | \nu) || P(\mathbf{U} | \mathbf{O}, \varphi))$ becomes the maximization of the bound on the RHS given the model parameters.

Using the original variable notation used in Figure 3, the bound can be expanded as follows:

$$\begin{aligned} &\sum_{k=1}^{\infty} \{E_Q[\log P(\pi_k | \alpha)] + E_Q[\log P(\theta_k^C | \mathbf{G}_0^C)] + E_Q[\log P(\theta_k^T | \mathbf{G}_0^T)]\} \\ &+ \sum_{n=1}^N \{E_Q[\log P(Z_n | \pi_1, \pi_2, \dots)] + E_Q[\log P(C_n | Z_n, \theta_1^C, \theta_2^C, \dots)] \\ &+ E_Q[\log P(T_n | Z_n, \theta_1^T, \theta_2^T, \dots)] + E_Q[\log P(L_n | T_n, \theta_s^L)]\} \\ &- E_Q[\log Q(\mathbf{T}, \mathbf{Z}, \boldsymbol{\theta}^C, \boldsymbol{\theta}^T, \boldsymbol{\pi} | \nu)] \end{aligned} \quad (5)$$

where \mathbf{T} , \mathbf{Z} , $\boldsymbol{\theta}^C$, $\boldsymbol{\theta}^T$, and $\boldsymbol{\pi}$ represent the collection of variables T_n , Z_n , θ_k^C , θ_k^T , and π_k respectively.

As described in Section 2.2, $P(\pi_k | \alpha)$ can be set to the Beta distribution $\mathbf{Beta}(1, \alpha)$. In our framework, we model the content information of text fragments as a mixture model of tokens in the set of vocabularies V . Hence $P^C(\cdot | \theta_k^C)$ is a multinomial distribution and \mathbf{G}_0^C can be defined as the Dirichlet distribution $G_0^C(\cdot | \boldsymbol{\mu}^C)$ where $\boldsymbol{\mu}^C$ is the set of hyper parameters. T_n follows the binomial distribution $P^T(\cdot | \theta_k^T)$ and hence \mathbf{G}_0^T is the Beta distribution $G_0^T(\cdot | \boldsymbol{\mu}^T)$ where $\boldsymbol{\mu}^T$ is the hyper parameter. The layout information is modeled by a set of Bernoulli trials, denoted as F_s . The outcome of each Bernoulli trial is whether the underlying text fragment possesses the f_s -th formatting feature in page s depending

on the value of θ_s^L and T_n , where $1 \leq f_s \leq |F_s|$. Therefore, $P(L_n | T_n, \theta_s^L)$ is represented by a set of binomial distributions. According to stick-breaking process, we can express: $P(Z_n | \pi_1, \pi_2, \dots) = \prod_{i=1}^{\infty} (1 - \pi_i)^{X_{Z_n > i}} \pi_i^{X_{Z_n = i}}$. Next, we make use of the truncated stick-breaking process [6] and define $Q(\mathbf{T}, \mathbf{Z}, \boldsymbol{\theta}^C, \boldsymbol{\theta}^T, \boldsymbol{\pi})$ as follows:

$$Q(\mathbf{T}, \mathbf{Z}, \boldsymbol{\theta}^C, \boldsymbol{\theta}^T, \boldsymbol{\pi}) = \prod_{k=1}^{K-1} Q_{\pi}(\pi_k | \tau_{k,0}, \tau_{k,1}) \prod_{k=1}^K Q_{\theta^T}(\theta_k^T | \delta_{k,0}, \delta_{k,1}) \prod_{k=1}^K Q_{\theta^C}(\theta_k^C | \zeta) \prod_{n=1}^N Q_T(T_n | \omega_n) \prod_{n=1}^N Q_Z(Z_n | \phi_n) \quad (6)$$

where K is the truncation level; $Q_{\pi}(\cdot | \tau_{k,0}, \tau_{k,1})$ is the Beta distribution $\mathbf{Beta}(\tau_{k,0}, \tau_{k,1})$; $Q_{\theta^T}(\theta_k^T | \delta_{k,0}, \delta_{k,1})$ is the Beta distribution $\mathbf{Beta}(\delta_{k,0}, \delta_{k,1})$; $Q_{\theta^C}(\theta_k^C | \zeta)$ is the Dirichlet distribution with parameter set ζ ; $Q_T(T_n | \omega_n)$ is the binomial distribution with parameter ω_n ; and $Q_Z(Z_n | \phi_n)$ is the multinomial distribution with parameter set $\phi_{n,1}, \dots, \phi_{n,K}$. In the truncated stick breaking process, $Q_Z(Z_n | \phi_n) = 0$ for $Z_n > K$. Under this setting, all the terms in Equation 5 can be expressed in explicit form.

To maximize Equation 5, we can take the first derivative with respect to each of the variational variables and set to 0. Next we obtain the following optimal conditions:

$$\begin{aligned} \tau_{k,0} &= (1 - \alpha) + \sum_{n=1}^N \phi_{n,k} \\ \tau_{k,1} &= \alpha + \sum_{n=1}^N \sum_{j=k+1}^K \phi_{n,j} \end{aligned} \quad (7)$$

$$\begin{aligned} \delta_{k,0} &= \mu_0^T + \sum_{n=1}^N \omega_n \phi_{n,k} \\ \delta_{k,1} &= \mu_1^T + \sum_{n=1}^N (1 - \omega_n) \phi_{n,k} \end{aligned} \quad (8)$$

$$\zeta_{k,j} = \mu_j^C + \sum_{n=1}^N w_{n,j} \phi_{n,k} \quad (9)$$

$$\begin{aligned} \phi_{n,k} \propto &\exp \{ \sum_{j=1}^{K-1} [\Psi(\tau_{j,1}) - \Psi(\tau_{j,0} + \tau_{j,1})] \\ &+ \Psi(\tau_{k,0}) - \Psi(\tau_{k,0} + \tau_{k,1}) \\ &+ \sum_{j=1}^{|V|} w_{n,j} [\Psi(\zeta_{k,j}) - \Psi(\sum_{h=1}^{|V|} \zeta_{k,h})] \\ &+ \omega_n (\Psi(\delta_{k,0}) - \Psi(\delta_{k,0} + \delta_{k,1})) \\ &+ (1 - \omega_n) (\Psi(\delta_{k,1}) - \Psi(\delta_{k,0} + \delta_{k,1})) \} \end{aligned} \quad (10)$$

where $w_{n,j} = 1$ if text fragment x_n contains j -th token in the vocabulary V , and 0 otherwise; $\Psi(\gamma)$, which is called digamma function, is the first derivative of the log Gamma function.

$$\omega_n = \frac{1}{1 + e^{-h(\phi_{n,k}, \delta_{k,0}, \delta_{k,1}, \theta_s^L)}} \quad (11)$$

where

$$\begin{aligned} &h(\phi_{n,k}, \delta_{k,0}, \delta_{k,1}, \theta_s^L) \\ &= \sum_{k=1}^K \phi_{n,k} (\Psi(\delta_{k,0}) - \Psi(\delta_{k,1})) + \sum_{l=1}^L u_{n,l} (\log \theta_{s,l}^L - \log(1 - \theta_{s,l}^L)) \end{aligned}$$

and $u_{n,f} = 1$ if x_n contains the f -th layout format in F_s in page $s = s(x_n)$, and 0 otherwise. Given the model parameters, one can then apply the steepest ascent algorithm, which is an iterative algorithm to update each variable at a time, until convergence.

Essentially, the attribute that x_n belongs to can be decided by the values of $\phi_{n,k}$ for $k = 1, 2, \dots$, each of which represents how likely that x_n is generated from the k -th mixture component. It can be observed that the value of $\phi_{n,k}$ depends on three different aspects in Equation 10. The first aspect is the prior proportion of the k -th components, which is characterized by the value of the variational parameters $\tau_{k,0}$ and $\tau_{k,1}$. The second aspect is the content of x_n , which is denoted by $w_{n,j}$, and the token distribution in k -th component, which is characterized by $\zeta_{k,j}$. The third aspect is likelihood that x_n belongs to an attribute, which is characterized by ω_n , and the prior distribution that a text fragment longs to an attribute value in the k -th component, which is characterized by $\delta_{k,0}$ and $\delta_{k,1}$. On the other hand, the probability that x_n is an attribute value is represented by the value of ω_n , which depends on other three aspects. The first aspect is $\phi_{n,k}$ which is the probability

that x_n belongs to the k -th component. The second aspect is the prior information about how likely a text fragment in the k -th component is an attribute value, characterized by the factor $\Psi(\delta_{k,0}) - \Psi(\delta_{k,1})$. The third aspect is the layout of x_n , which is characterized by $f_{n,l}$, and the factor about the layout format of an attribute, which is characterized by $(\log \theta_{s,f}^L - \log(1 - \theta_{s,f}^L))$. Interestingly, it shows that both normalization and extraction decision have mutual influence in the optimal condition according to Equations 10 and 11. In particular, Equation 11 is in the form of logistic regression, which is discriminative in nature, considering a factor related to the k -th component in the mixture, as well as the layout format of the text fragment. Consequently, our model can resolve the conflict between the extraction and normalization tasks and achieve an optimal solution.

3.2 Unsupervised Approach

As described in the previous section, we can automatically achieve the optimal extraction and normalization of product attributes by satisfying the optimal condition stated in Equations 7-11 given the model parameters. We have developed a method which can automatically determine the model parameters and initialize a steepest ascent algorithm, achieving an unsupervised extraction and normalization.

As exemplified in Section 1.1, our framework can consider the page-dependent layout format of text fragments to enhance extraction. However, the layout information of an unseen Web page is unknown and hence we cannot predefine or estimate the values of $\theta_{s,f}^L$. As a result, we develop an Expectation-Maximization (EM) algorithm based on our variational method to estimate the values of all $\theta_{s,f}^T$ in page s . By taking the first derivative of Equation 5 with respect to each $\theta_{s,f}^L$, we can obtain the following formula:

$$\theta_{s,f}^F = \frac{1}{NK} \sum_{n=1}^N \sum_{k=1}^K \omega_n u_{n,f} \quad (12)$$

This is the optimal value of each $\theta_{s,f}^L$ given other model parameters and the current set of variational parameters. The E-step and M-step are defined as follows:

E-step:

Apply steepest ascent algorithm until convergence to achieve the optimal conditions depicted in Equations 7-11.

M-step:

Calculate each $\theta_{s,f}^T$ using Equation 12.

To initialize the EM algorithm, we are required to estimate $\phi_{n,k}$ and ω_n for the n -th text fragment. To achieve this, we make use of the prior knowledge, which is in the form of a list of a few terms, denoted as κ , related to product attributes. Let κ_i be the i -th term in the list. Notice that the terms are not required to be categorized into different attributes. In practice, this list can be easily obtained, for example, by scanning one Web page containing a product in the underlying domain and highlighting the product attributes. For each κ_i , we select the i -th component in our model and set a higher value of $\zeta_{i,j}$ if κ_i is equal to the $v_j \in V$, and zero otherwise. In particular, we set to 10 for such $\zeta_{k,j}$. Next, for these components, we set $\delta_{i,0} = 6$ and $\delta_{i,1} = 4$ which essentially means that 6 out of 10 text fragments in this component will be a text fragment related to attribute values. $\delta_{k,0}$ and $\delta_{k,1}$ are set to 4 and 6 respectively for other components. ω_n can then be calculated according to Equation 11. Notice that these values are used in initialization only. Updated values will be automatically calculated via the EM algorithm.

For the model parameters, α is the scaling parameter between 0 and 1 in the Dirichlet process, which essentially af-

fects the number of normalized attributes in the normalization process. Since we apply our framework to the domains, for example, digital camera, in which each product contains a number of attributes, we set α to a value that favors large number of normalized attributes. In particular we set α to 0.1. μ_j^T refers to the prior knowledge about how likely a text fragment will be an attribute value. We treat it as an uninformative prior and set $\mu_0^T = \mu_1^T = 1$. Similarly, μ_j^C are treated as uninformative as all μ_j^C are set to 1. The truncation level K of truncated stick breaking process is set to a relatively large value such that attributes are normalized as fine as possible. In particular, K is set to 500.

4. EXPERIMENTAL RESULTS

4.1 Experiment Setup

We have conducted extensive experiments on four different domains, namely, digital camera, MP3 player, camcorder, and restaurant domains to evaluate our framework. We have collected 85 Web pages from 41 sites, 96 Web pages from 62 sites, and 111 Web pages from 61 sites in the digital camera, MP3 player, and camcorder domain respectively. In these three domains, each Web page contains one product and a number of product attributes. The dataset of the restaurant domain contains 29 Web pages from the LA-weekly Restaurant Guide¹. Each page contains attributes including names, addresses, phone numbers, and review from customers for one or more restaurants. We use a simple method which considers the line separators such as the HTML tags $\langle \text{BR} \rangle$, $\langle \text{P} \rangle$, $\langle \text{LI} \rangle$, etc. to collect all text fragments from all Web pages. The layout format of each text fragment is automatically recorded during the collection process. For evaluation purpose, two human accessors were invited to annotate each text fragment by indicating whether it is a valid product attribute value and normalize all identified attribute values to appropriate reference attributes. If there is a disagreement on the judgment of the two human accessors, it is resolved by a discussion among them.

In each domain, we conducted 10 runs of experiments. For each run, we randomly selected one page in the domain and selected the text fragments corresponding to attribute values. The tokens in these text fragments were used to initialize our algorithm as stated in Section 3.2. In practice, this could simply be done in Web browser by highlighting the portion containing attribute values. Next, we applied our framework to all other remaining pages to simultaneously extract and normalize product attributes. The performance of both extraction and normalization in each run were recorded to evaluate our framework.

4.2 Evaluation on Product Attribute Normalization

We evaluate the performance of product attribute normalization. The attribute normalization results are compared with the manually annotated answers. We adopt the pairwise precision and recall, which are commonly used in clustering, as the evaluation metric. Pairwise recall is defined as the number of pairs of text fragments, which are correctly predicted as referring to the same reference attribute by the system, divided by the actual number of pairs of text fragments referring to the same reference attribute. Pairwise precision is defined as the number of pairs of text fragments,

¹The restaurant dataset is available for download in the URL: <http://www.isi.edu/info-agents/RISE/>.

	DC			MP3			CAM		
	P	R	F	P	R	F	P	R	F
1	0.84	0.90	0.87	0.80	0.76	0.78	0.87	0.86	0.87
	(0.64)	(0.19)	(0.30)	(0.63)	(0.18)	(0.28)	(0.60)	(0.23)	(0.33)
2	0.79	0.87	0.83	0.75	0.72	0.74	0.87	0.84	0.86
	(0.54)	(0.18)	(0.27)	(0.56)	(0.17)	(0.26)	(0.55)	(0.26)	(0.35)
3	0.70	0.76	0.73	0.76	0.75	0.76	0.84	0.83	0.83
	(0.59)	(0.20)	(0.30)	(0.66)	(0.23)	(0.34)	(0.52)	(0.23)	(0.31)
4	0.75	0.72	0.74	0.67	0.72	0.69	0.82	0.86	0.84
	(0.49)	(0.15)	(0.23)	(0.55)	(0.23)	(0.33)	(0.51)	(0.22)	(0.31)
5	0.80	0.74	0.77	0.62	0.70	0.66	0.83	0.86	0.85
	(0.51)	(0.16)	(0.25)	(0.45)	(0.20)	(0.28)	(0.58)	(0.23)	(0.33)
6	0.74	0.81	0.77	0.71	0.68	0.69	0.77	0.83	0.80
	(0.55)	(0.19)	(0.28)	(0.40)	(0.26)	(0.32)	(0.54)	(0.24)	(0.34)
7	0.74	0.70	0.72	0.69	0.77	0.73	0.80	0.81	0.81
	(0.42)	(0.13)	(0.20)	(0.40)	(0.29)	(0.34)	(0.47)	(0.21)	(0.29)
8	0.70	0.79	0.74	0.45	0.54	0.49	0.77	0.75	0.76
	(0.55)	(0.20)	(0.29)	(0.39)	(0.27)	(0.32)	(0.44)	(0.18)	(0.25)
9	0.78	0.83	0.80	0.59	0.62	0.60	0.79	0.78	0.79
	(0.53)	(0.16)	(0.25)	(0.42)	(0.29)	(0.34)	(0.49)	(0.18)	(0.27)
10	0.79	0.80	0.79	0.61	0.60	0.61	0.64	0.71	0.67
	(0.55)	(0.18)	(0.27)	(0.36)	(0.27)	(0.31)	(0.47)	(0.19)	(0.27)
Avg.	0.76	0.79	0.78	0.67	0.69	0.68	0.80	0.81	0.81
	(0.54)	(0.18)	(0.26)	(0.48)	(0.24)	(0.31)	(0.52)	(0.22)	(0.31)

Table 1: The attribute normalization performance on the digital camera (DC), MP3 player (MP3), and camcorder (CAM) domains. P, R, and F refer to the pairwise recall, precision, and F_1 -measure respectively. Avg. refers to the average performance.

which are correctly predicted as referring to the same reference attribute by the system, divided by the total number of pairs of text fragments, which are predicted as referring to the same reference attribute. Pairwise F_1 -measure is defined as the harmonic mean of equal weighting of pairwise recall and precision.

We first collected all correctly extracted text fragments by our framework in each run of our experiments. The pairwise precision, recall, and F_1 -measure of the product attributes were then calculated. We conducted the evaluation for the digital camera, MP3 player, and camcorder domains because the products in these domain contain a large number of attributes, and some of these attributes are previously unseen. The restaurant domain is not evaluated since each restaurant only consists of a few attributes including names, phone number, addresses, customer reviewers, and credit card information which can be easily predefined. We design a baseline approach for comparison. For each pair of the text fragments correctly extracted by our framework, we compute the edit-distance as described in [2]. Attribute normalization is then conducted by invoking the agglomerative clustering. This baseline approach only considers the text content of text fragments.

Table 1 shows the attribute normalization performance of our framework and the baseline approach. Each row of the table corresponds to a run of the experiment and the last row is the average performance. Each cell records the performance of our framework and the performance of the baseline approach is shown in brackets. Each column refers to the extraction performance in a domain. Our framework achieves a better results compared with the baseline approach. In particular, the average F_1 -measure are 0.78, 0.68, and 0.81 in the digital camera, MP3 player, and camcorder domains respectively. It shows that our framework can effectively normalize text fragments with similar semantic meaning to the same reference product attribute. The baseline approach has a relatively low recall since it can only consider the token content of the text fragments. In contrast, each mixture component of our framework has its own distribution of terms, so that tokens related to the reference attributes will also be considered. Therefore, our framework can normalize attributes with no common token between text fragments

Att. 1	Att. 2	Att. 3	Att. 4	Att. 5
memory card secure digital flash	zoom optic digital lens megapixel	battery rechargeable include alkaline lithium	dimension height width width inch	flash auto reduction built-in mode
Att. 6	Att. 7	Att. 8	Att. 9	Att. 10
megapixel resolution effective total digital	mode shoot scene flash portrait	movie mode shoot sound audio	lens type system optic aperture	focus auto mode range macro

Table 2: The visualization of the top five weighted terms in the ten largest normalized attributes in the digital camera domain.

such as “Night Portrait” and “Candle Light”, which refers to the reference attribute “shooting mode” of a digital camera. Table 2 shows the top 5 weighted terms in 10 largest normalized attributes in the digital camera domain. It can be observed that the semantic meaning of the attributes can be easily interpreted from the terms. The output of attribute normalization can be very useful for supporting other intelligent applications such as product attribute indexing and product retrieval.

4.3 Evaluation on Product Attribute Extraction

We evaluate the extraction performance of our framework in the digital camera, MP3 player, camcorder, and restaurant domains. The system extracted attributes are compared with the attributes extracted by human as described above. We adopt the commonly used recall and precision as the evaluation metrics. Recall is defined as the number of correctly extracted text fragments corresponding to attribute values divided by the actual number of text fragments corresponding to attribute values. Precision is defined as the number of correctly extracted text fragments corresponding to attribute values divided by the total number of text fragments extracted by the system. F_1 -measure defined as the harmonic mean of recall and precision is also used.

Table 3 shows the attribute extraction performance of our framework. Each row of the table depicts the extraction performance in a run. The last row shows the average extraction performance. Our approach obtains promising results in the four domains. The average F_1 -measure are 0.95, 0.69, 0.60, and 0.58 in the restaurant, digital camera, MP3 player, and camcorder domains respectively. Notice that our framework is an unsupervised approach and does not require human effort to prepare training examples for every Web site. Surprisingly, in the restaurant domain, our framework achieves a performance which is comparable to the supervised method stated in [9]. Moreover, our framework can extract product attributes reasonably well from over 300 Web pages which are originated from over 150 Web sites in the other three domains.

5. RELATED WORK

Various information extraction techniques have been proposed to extract attributes from semi-structured documents including Web pages [11, 18]. For example, Conditional Random Fields (CRF) [7] have been applied to extract information from Web documents achieving the state-of-the-art performance. Sarawagi and Cohen developed a semi-Markov CRF model which can assign labels to segments of a sequence [12]. Sutton et al. proposed a dynamic CRF models for labeling sequence data [14]. Zhu proposed an integrated model based on hierarchical CRF and semi-CRF for

	RES			DC			MP3			CAM		
	P	R	F	P	R	F	P	R	F	P	R	F
1	0.89	0.99	0.94	0.80	0.50	0.62	0.62	0.58	0.59	0.71	0.43	0.53
2	0.89	0.99	0.94	0.72	0.71	0.71	0.51	0.66	0.58	0.66	0.42	0.51
3	0.89	0.99	0.94	0.70	0.58	0.64	0.52	0.54	0.53	0.65	0.52	0.57
4	0.89	0.99	0.94	0.61	0.89	0.72	0.66	0.59	0.62	0.63	0.61	0.63
5	0.89	0.99	0.94	0.69	0.79	0.74	0.53	0.68	0.59	0.66	0.55	0.60
6	0.89	1.00	0.94	0.73	0.59	0.65	0.56	0.68	0.62	0.64	0.52	0.57
7	0.91	0.99	0.95	0.60	0.90	0.72	0.63	0.57	0.60	0.53	0.67	0.59
8	0.92	0.99	0.96	0.81	0.63	0.71	0.53	0.69	0.60	0.57	0.83	0.68
9	0.93	0.99	0.96	0.74	0.65	0.70	0.60	0.60	0.60	0.61	0.57	0.59
10	0.99	1.00	0.99	0.74	0.61	0.67	0.53	0.77	0.63	0.59	0.46	0.52
Avg	0.91	0.99	0.95	0.71	0.69	0.69	0.57	0.64	0.60	0.63	0.58	0.58

Table 3: The attribute extraction performance on the restaurant (RES), digital camera (DC), MP3 player (MP3), and camcorder (CAM) domains. Avg. refers to the average performance.

detecting records and extracting attributes from raw Web pages [19]. However, one shortcoming of these supervised methods is that human effort is needed to prepare training examples. Moreover, the attributes to be extracted are pre-defined and hence it cannot discover unseen attributes. Wong and Lam aimed at reducing the human work of preparing training examples by automatically adapting extraction knowledge learned from a source Web site to new unseen sites and discover new attributes [17]. In this paper, we propose an unsupervised framework and thus no training example is needed. Probst et al. [10] proposed a semi-supervised algorithm to extract attribute value pairs from text description. Their approach aims at handling free text descriptions by making use of natural language processing techniques. Hence, it cannot be applied to Web documents which are composed of mixing HTML tags and free texts.

The objective of entity resolution shares certain resemblances with our goal of product attribute normalization. It aims at classifying whether two references refer to the same entity. Singla and Domingos developed an approach to entity resolution based on Markov Logic Network [13]. Bhattacharya and Getoor proposed an unsupervised approach for entity resolution based on Latent Dirichlet Allocation (LDA) [1]. One limitation of these approaches is that the entities are required to be extracted in advance and cannot be applied to raw data.

A common drawback of existing methods is that the extraction and normalization tasks are conducted in two separate steps, leading to conflict solutions and degrading overall performance. Approaches based on CRF have been proposed to collaboratively conduct information extraction and mining [8, 16]. However, conducting the attributes to be extracted have to be known in these approaches and previously unseen attributes cannot be handled.

Dirichlet process mixtures have been studied and applied in image analysis, language modeling [3, 15]. Our framework extend the Dirichlet process mixture model and shows that the mutual the content and layout information of text fragments can be considered jointly to achieve an optimal solution in product attribute extraction and normalization.

6. CONCLUSIONS

We have developed an unsupervised framework which aims at simultaneously extracting and normalizing product attributes from Web pages collected from different sites. Our method can effectively consider the page-independent content information and the page-dependent layout information of the text fragments of Web pages. We have developed a graphical model, which employs Dirichlet process prior, to model the generation of text fragments in Web pages. An unsupervised inference algorithm based on variational

method is derived. We formally show that content and layout information can collaborate and improve both extraction and normalization performance. Extensive experiments on four different domains have been conducted to show the robustness and effectiveness of our approach.

7. REFERENCES

- [1] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 47–58, 2006.
- [2] M. Bilenko and R. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 39–48, 2003.
- [3] D. Blei and M. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, 2006.
- [4] S.-L. Chuang, K. Chang, and C. Zhai. Context-aware wrapping: Synchronized data extraction. In *Proceedings of the Thirty-Third Very Large Databases Conference*, pages 699–710, 2007.
- [5] V. Crescenzi, G. Mecca, and P. Merialdo. ROADRUNNER: Towards automatic data extraction from large web sites. In *Proceedings of the Twenty-Seventh Very Large Databases Conference*, pages 109–118, 2001.
- [6] J. Ishwaran and L. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–174, 2001.
- [7] 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*, pages 282–289, 2001.
- [8] 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.
- [9] 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.
- [10] K. Probst, M. K. R. Ghai, A. Fano, and Y. Liu. Semi-supervised learning of attribute-value pairs from product descriptions. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 2838–2843, 2007.
- [11] J. Rurmo, A. Ageno, and N. Catala. Adaptive information extraction. *ACM Computing Surveys*, 38(2):Article 4, 2006.
- [12] S. Sarawagi and W. Cohen. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems 17, Neural Information Processing Systems*, 2004.
- [13] P. Singla and P. Domingos. Entity resolution with markov logic. In *Proceedings of the Sixth IEEE International Conference on Data Mining*, pages 572–582, 2006.
- [14] C. Sutton, K. Rohanimanesh, and A. McCallum. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of Twenty-First International Conference on Machine Learning*, pages 783–790, 2004.
- [15] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.
- [16] 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.
- [17] T.-L. Wong and W. Lam. Adapting web information extraction knowledge via mining site invariant and site dependent features. *ACM Transactions on Internet Technology*, 7(1):Article 6, 2007.
- [18] H. Zhao, W. Meng, and C. Yu. Mining templates from search result records of search engines. In *Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 884–892, 2007.
- [19] J. Zhu, B. Zhang, Z. Nie, J.-R. Wen, and H.-W. Hon. Webpage understanding: an integrated approach. In *Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 903–912, 2007.