Normalizing Web Product Attributes and Discovering Domain Ontology with Minimal Effort *

Tak-Lam Wong
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
wongtl@cse.cuhk.edu.hk

Lidong Bing Wai Lam
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Shatin, Hong Kong
{ldbing, wlam}@se.cuhk.edu.hk

ABSTRACT

We have developed a framework aiming at normalizing product attributes from Web pages collected from different Web sites without the need of labeled training examples. It can deal with pages composed of different layout format and content in an unsupervised manner. As a result, it can handle a variety of different domains with minimal effort. Our model is based on a generative probabilistic graphical model incorporated with Hidden Markov Models (HMM) considering both attribute names and attribute values to extract and normalize text fragments from Web pages in a unified manner. Dirichlet Process is employed to handle the unlimited number of attributes in a domain. An unsupervised inference method is proposed to predict the unobservable variables. We have also developed a method to automatically construct a domain ontology using the normalized product attributes which are the output of the inference on the graphical model. We have conducted extensive experiments and compared with existing works using product Web pages collected from real-world Web sites in three different domains to demonstrate the effectiveness of our framework.

Categories and Subject Descriptors

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

General Terms

Algorithms

*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 Codes: CUHK4128/07 and CUHK413510) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050442 and 2050476). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

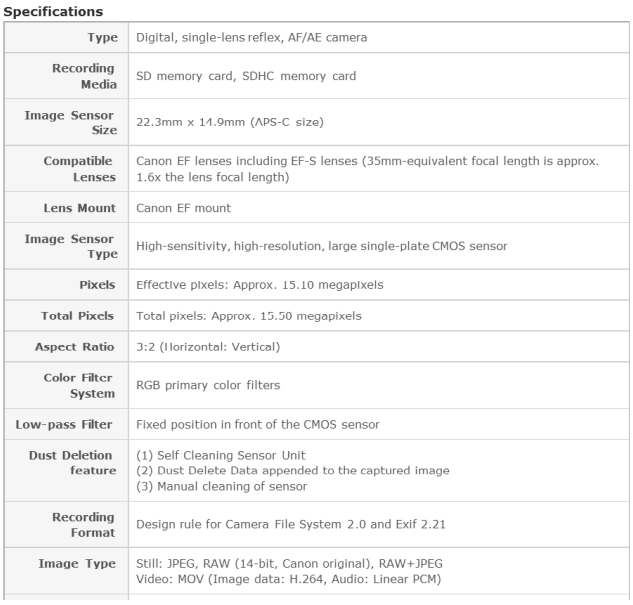
WSDM'11, February 9–12, 2011, Hong Kong, China.

Keywords

information extraction, graphical models, Web mining

1. INTRODUCTION

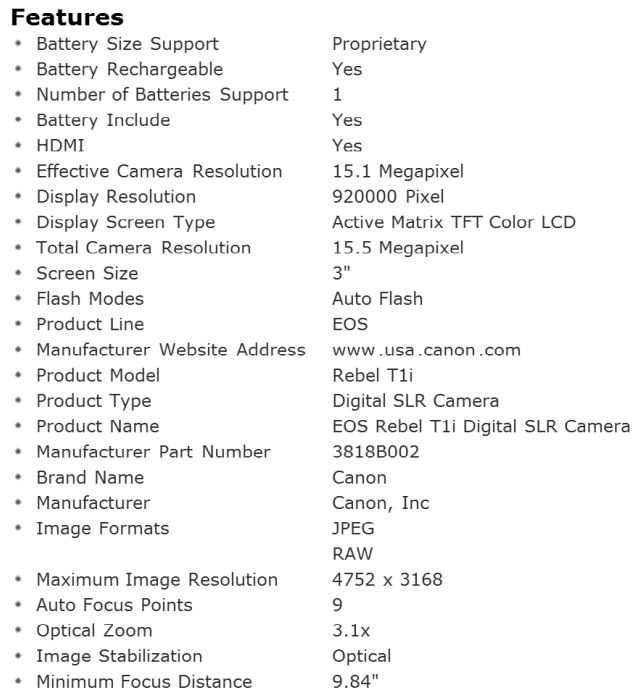
The World Wide Web (WWW) consists of a massive amount of Web pages about different products, each of which is described by a number of attributes. For example, Figure 1 shows a Web page containing a digital camera collected from a retailer Web site. The digital camera consists of product attributes like *type*, *recording media*, etc, displayed in a tabular form. The first column and the second column of the table refer to the *attribute name* and the corresponding *attribute value* respectively. Users can learn the characteristic of the product by manually browsing the Web page and identify the attributes. Unfortunately, Web pages are typically written by different retailer Web sites and are likely to convey incomplete information using different formats or terminology. For instance, Figure 2 shows another Web page containing the same digital camera, but collected from a retailer Web site different from the one shown in Figure 1. The product attributes are displayed as list items. Both Figures 1 and 2 contain different parts of the attributes of the product. Even referring to the same attribute, say, *image format*, the attribute name and the attribute value in Figure 1 are the text fragments "Image Type" and "Still: JPEG, RAW (14-bit, Canon original), RAW+JPEG" respectively, while the attribute name and the attribute value in Figure 2 are the text fragments "Image Formats" and "JPEG RAW" respectively. In essence, attribute name and attribute value embody different types of information. Attribute name mainly embodies the semantic meaning of the text fragment, while attribute value mainly embodies the characteristic of the product. Though these text fragments from these two figures consist of some common tokens in both attribute name and attribute value, it is not straightforward to infer that the text fragments describe the same attribute. Notice that attribute names sometimes are missing in some of the Web pages, imposing more difficulty in inferring the attribute of a text fragment. Moreover, common tokens become even rarer when two Web pages contain different products, whose attribute values are likely to be different. As a consequence, it is difficult for a user to obtain complete product attribute information about a product as he/she needs to manually browse multiple Web pages in different formats, extract the attribute information, and identify text fragments referring to the same attribute to filter the redundant information.



Specifications

Type	Digital, single-lens reflex, AF/AE camera
Recording Media	SD memory card, SDHC memory card
Image Sensor Size	22.3mm x 14.9mm (APS-C size)
Compatible Lenses	Canon EF lenses including EF-S lenses (35mm-equivalent focal length is approx. 1.6x the lens focal length)
Lens Mount	Canon EF mount
Image Sensor Type	High-sensitivity, high-resolution, large single-plate CMOS sensor
Pixels	Effective pixels: Approx. 15.10 megapixels
Total Pixels	Total pixels: Approx. 15.50 megapixels
Aspect Ratio	3:2 (Horizontal: Vertical)
Color Filter System	RGB primary color filters
Low-pass Filter	Fixed position in front of the CMOS sensor
Dust Deletion feature	(1) Self Cleaning Sensor Unit (2) Dust Delete Data appended to the captured image (3) Manual cleaning of sensor
Recording Format	Design rule for Camera File System 2.0 and Exif 2.21
Image Type	Still: JPEG, RAW (14-bit, Canon original), RAW+JPEG Video: MOV (Image data: H.264, Audio: Linear PCM)

Figure 1: A sample of a portion of a Web page containing a digital camera collected from a Web site. (Web site URL: *www.adorama.com*)



Features

- Battery Size Support: Proprietary
- Battery Rechargeable: Yes
- Number of Batteries Support: 1
- Battery Include: Yes
- HDMI: Yes
- Effective Camera Resolution: 15.1 Megapixel
- Display Resolution: 920000 Pixel
- Display Screen Type: Active Matrix TFT Color LCD
- Total Camera Resolution: 15.5 Megapixel
- Screen Size: 3"
- Flash Modes: Auto Flash
- Product Line: EOS
- Manufacturer Website Address: www.usa.canon.com
- Product Model: Rebel T1i
- Product Type: Digital SLR Camera
- Product Name: EOS Rebel T1i Digital SLR Camera
- Manufacturer Part Number: 3818B002
- Brand Name: Canon
- Manufacturer: Canon, Inc
- Image Formats: JPEG, RAW
- Maximum Image Resolution: 4752 x 3168
- Auto Focus Points: 9
- Optical Zoom: 3.1x
- Image Stabilization: Optical
- Minimum Focus Distance: 9.84"

Figure 2: A sample of a portion of a Web page containing the same digital camera shown in Figure 1, but collected from another Web site. (Web site URL: *www.buy.com*)

A few approaches have been proposed to address the product attribute normalization problem whose objective is to automatically discover the underlying product attributes. Guo et al. proposed a method called latent semantic association (LaSA) to categorize product features extracted from opinion presented in certain format, sharing certain similarity with product attribute normalization [8]. They first identify product feature candidates from the *Pros* and *Cons* columns in Web opinions and form virtual context document for each of them. LaSA is then applied to the product feature candidates to learn the latent relationship among the terms related to product features. Next, another level of LaSA is employed to categorize the product features. One limitation of their approach is that they focus on opinions presented in certain fixed formats. Moreover, they aim at identifying the product features, ignoring the feature values of the product.

Numerous approaches, such as information extraction wrapper, have been proposed to extract information from Web pages [16, 4]. The existing approaches can be categorized into supervised and unsupervised learning methods. One major limitation of supervised information extraction methods is that they require human effort to prepare a substantial amount of training examples to train the extraction model. Another limitation is that the attributes of interest must be specified in advance. For example, one has to define that a digital camera contains three attributes *resolution*, *file format*, and *memory storage*. The learned information extraction model can only extract these three attributes. Other unspecified or previously unseen attributes cannot be extracted. Unsupervised learning methods attempt to tackle these problems. However, the semantic meaning of the extracted information is unknown. For example, one may not know the semantic meaning of the extracted text fragment "High-sensitivity, high-resolution, large single-plate CMOS sensor" in Figure 1 is the attribute value for the attribute *Image Sensor Type* without any domain knowledge. To solve this problem, some approaches have been proposed to extract the attribute names and attribute values [12, 19, 20]. However, they can only handle some fixed layout format Web pages. Moreover, these approaches treat different attribute names as different attributes. However, as mentioned above, attribute names for the same attribute can be different if they are extracted from different Web sites.

In this paper, we have developed a framework for automatically normalizing Web product attributes, which aims at identifying text fragments of Web pages referring to the same attribute, without the need of labeled training examples. Our framework can deal with Web pages originated from different Web sites and formatted in different styles. Therefore, it can handle different domains with minimal human effort. Our framework is designed based on generative probabilistic graphical models incorporated with Hidden Markov Models (HMM). One characteristic of our model is that it considers the attribute names and the attribute values of the products described in Web pages improving the normalization of text fragments to appropriate attributes. Another characteristic is that Dirichlet Process is employed to handle unlimited number of attributes, which do not need to be predefined in advance. We have also developed an application by utilizing the extracted and normalized Web product attributes to construct a domain ontology in an automatic manner. For example, Figure 3 depicts a part of the ontology of the digital camera domain generated in our experiment. Each node, which is associated with a set of terms, in the ontology represents a concept of the domain. Such domain ontology can be utilized in other differ-

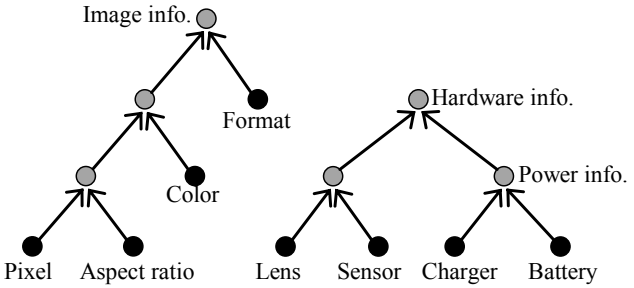


Figure 3: Parts of the ontology in the digital camera domain.

ent intelligent tasks such as conducting inference. We have conducted extensive experiments on the Web pages of three products collected from real-world Web sites to demonstrate the effectiveness of our framework.

We have previously investigated the problem of extracting and normalizing product attributes and reported in [18]. The framework proposed in this paper is largely different from the previous work in several aspects. The model proposed in this paper is more sophisticated considering attribute names and attribute values of the text fragments to improve the performance in attribute normalization. In light of this, the variational inference method reported in [18] is not applicable and a new inference approach is proposed in this paper. Moreover, this paper also presents a method for automatically constructing the domain ontology from the extracted and normalized product attributes.

2. PROBLEM DEFINITION

In a product domain $\mathcal{D}$, let $\mathcal{A}$ denote a set of reference attributes and a_i be the i-th attribute in $\mathcal{A}$. For example, in the digital camera domain, reference attributes of digital cameras may include "lcd-screen-size", "effective-pixels", "focal-length", etc. We design 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 ∞.

Given a collection of product record Web pages $\mathcal{W}$ collected from a set of Web sites $\mathcal{S}$. Let $w_i(s)$ be i-th page collected from the site s. Within the Web page $w_i(s)$, we can collect a set of text fragments $X(w_i(s))$. For example, "Type Digital, single-lens reflex, AF/AE camera" and "Image Type Still: JPEG, RAW (14-bit, Canon original), RAW+JPEG" are samples of text fragments collected from the page shown in Figure 1. Let $x_j(w_i(s))$ be the j-th text fragment in the Web page $w_i(s)$. Essentially, each x in $X(w_i(s))$ can be represented by a five-field tuple (U, Q, L, T, A). U refers to the tokens of each text fragment, and Q refers to the label information of the tokens, i.e., q_1 represents the attribute name information, labeled as "attribute-name", and q_2 represents the attribute value information contained in the text fragment, labeled as "attribute-value", respectively. In particular, $\bar{q}$ represents that the token is a "attribute-irrelevant" token. Take the fragment "Image Type Still: JPEG, RAW (14-bit, Canon original), RAW+JPEG" as an example. The tokens "Image Type" refer to the attribute name, while the remaining tokens correspond to the attribute value. For another example, the text fragment "Specifications" corresponds to neither the attribute name nor the attribute value, so it refers to attribute-irrelevant information. L refers to the layout information of the text fragment. For example, the text fragment "Specifications" is in boldface and in larger font size in Figure 1. T, defined as the target information, is a binary variable which is equal to 1 if the underlying text fragment is related to an attribute in $\mathcal{A}$, and 0 otherwise. For example, the values of T for the text fragments "Specifications" and "Image Type Still: JPEG, RAW (14-bit, Canon original), RAW+JPEG" 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 the elements in $\mathcal{A}$. For example, the values of A for the text fragments "Image Type Still: JPEG, RAW (14-bit, Canon original), RAW+JPEG" and "Image Formats JPEG RAW" collected from Figures 1 and 2 respectively should be equal to the reference attribute "image format" included in $\mathcal{A}$.

In practice, the layout information L and the token information U of a text fragment can be observed from Web pages. However, the target information T, the attribute information A and the label information of tokens Q cannot be observed. As a result, the task of attribute normalization can be defined as the prediction of the value of A for each text fragment, so that one can obtain the reference attribute to which the underlying text fragment refers. Formally, for each text fragment, we aim at finding $A = a^*$, such that

$$a^* = \arg\max_a P(A = a|L, U) \tag{1}$$

Meanwhile, our framework predicts the label information of tokens Q for each text fragment, and the information can help with the task of extraction as well as the task of normalization. Formally, for each text fragment, we aim at finding $Q = q^*$, such that

$$q^* = \arg\max_q P(Q = q|L, U) \tag{2}$$

When $T = 1$, we have $P(A = a|L, U) > 0$, $P(Q = q, q \in q_1, q_2|L, U) > 0$ for some $a \in \mathcal{A}\backslash\bar{a}$ and $P(A = \bar{a}|L, U) = 0$. When $T = 0$, we have $P(A = \bar{a}|L, U) = 1$, $P(Q = \bar{q}|L, U) = 1$. 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, A and Q such that the joint probability $P(T, A, Q|L, U)$ can be maximized leading to a solution satisfying both tasks.

3. OUR MODEL

Our proposed framework is based on a specially designed graphical model as depicted in Figure 4. 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. Table 1 illustrates the meaning of each notation used in our framework.

We employ Dirichlet process prior to tackle our problem. Each mixture component refers to a reference attribute in our framework. As a result, our framework can handle unlimited number of reference attributes. Essentially, our framework can be viewed as a mixture model containing unlimited number of components with different proportion. Each component refers to a reference attribute in the domain. Suppose we have a collection of N different text frag-

Symbol	Meaning
N	The number of text fragments
S	The number of Web sites
x_n	The n-th text fragment
M_n	The number of tokens of the n-th text fragment
Z_n	The "index" of the parameters, stating the cluster from which the text fragment comes
U_n	The tokens information $U_{n,1}, U_{n,2}, \ldots, U_{n,M_n}$ of the n-th text fragment
Q_n	The label information of tokens $Q_{n,1}, Q_{n,2}, \ldots, Q_{n,M_n}$
W_n	The set of tokens $U_{n,1}, U_{n,2}, \ldots, U_{n,M_n}$ and labels $Q_{n,1}, Q_{n,2}, \ldots, Q_{n,M_n}$ of the n-th text fragment
T_n	The target variable illustrating whether the text fragment is related to product attribute
L_n	The layout of the n-th text fragment, indicating whether it has or has not some particular layout
π_k	The proportion of the kth component in the mixture
θ_k^T	A set of binomial distribution parameters for generating T_n
θ_k^H	The set of parameters of the k-th HMM model
θ_s^L	A set of site-dependent parameters controlling the layout format of each text fragment on the page
α	The parameter denoted in the stick breaking of Dirichlet process
G_0^T	The hyper parameter, or prior process to generate θ_k^T
G_0^H	The hyper parameter, or prior process to generate θ_k^H

Table 1: Notations Used in Our Framework

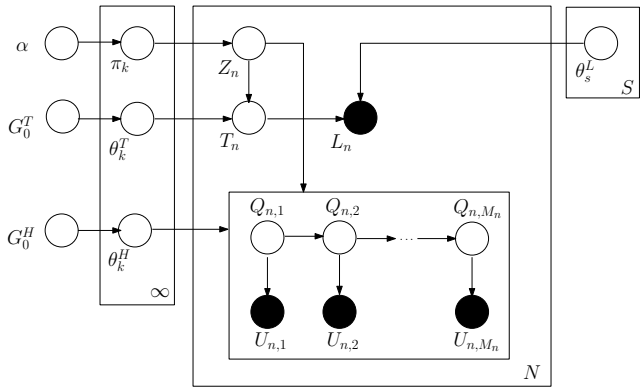


Figure 4: The graphical model for the generation of text fragments in Web pages

ments collected from S different Web sites. A text fragment refers to the text unit displayed in Web browser and can be identified by considering the Document Object Model (DOM)[1] structure of a Web page. 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 representing the index of the mixture component from which the underlying text fragment is generated. Essentially, A_n is replaced with Z_n for clarity and $A_n = a_{z_n}$ where $a_i \in \mathcal{A}$. We also employ Hidden Markov Models (HMM) to predict the label of each token of the N text fragments. As mentioned in Section 2, we use three kinds of labels, namely, "attribute-name", "attribute-value" and "attribute-irrelevant". Suppose there are M_n tokens of the n-th text fragment. We assume that each mixture component consists of an individual HMM. Hence through the variable Z_n we can find the corresponding HMM of the n-th text fragment for labeling its tokens. The token information U_n, also known as the page-independent content information, is then generated according to $P^H(U_n|Q_n, Z_n, \theta_k^H)$, where $P^H(\cdot|Q_n, Z_n, \theta_k^H)$ is the probability distribution about the token information U_n given the variables Q_n, Z_n and θ_k^H. U_n represents the sequence of tokens $U_{n,1}, U_{n,2}, \ldots, U_{n,M_n}$, while Q_n represents the label information of tokens $Q_{n,1}, Q_{n,2}, \ldots, Q_{n,M_n}$. θ_k^H refers to the set of parameters of the k-th HMM model.

Next, 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 page-dependent layout format of the text fragments in the page s. There is mutual cooperation between the layout information and the target information of a text fragment. T_n together with θ_s^L will generate the page-dependent 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.

In ordinary Dirichlet mixture models, each mixture component consists of a distribution to characterize the data. Instead, our framework consists of two different distributions parameterized by θ_k^T and θ_k^H for the k-th component. θ_k^T and θ_k^H are in turn generated from the base distributions G_0^T and G_0^H respectively in the Dirichlet process. G_0^T and G_0^H act as the prior distributions of the target information and the component-relevant HMM information respectively. For example, suppose we model the target information of the text fragments. 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. Similarly, G_0^H can be a Dirichlet distribution which is the conjugate prior of a mixture model, $P^H(\cdot|\theta_k^H)$ is a multinomial distribution, and θ_k^H is the set of parameters of multinomial distribution in component k.

We adopt the stick breaking construction representation of Dirichlet process prior in the graphical model [15]. In summary, we can break a one-unit length stick for an infi-

[1] *http://www.w3.org/DOM/*

nite number of times. Each time, we break a π_k portion from the remaining portion of the stick according to $Beta(1, \alpha)$ in the k-th break, where $Beta(\alpha_1, \alpha_2)$ is the Beta distribution, with parameters α_1 and α_2. Therefore, the k-th piece of the broken sticks can represent the proportion of k-th component in the mixture. Dirichlet process prior can support an infinite number of mixture components, which refer to the reference attributes in our framework. Z_n is then drawn from the distribution π. In summary, the generation process can be described as follows:

$$\begin{array}{ll} \tilde{\pi_k}|\alpha \sim Beta(1,\alpha) & \pi_k = \tilde{\pi_k}\prod_{i=1}^{k-1}(1-\tilde{\pi_k}) \\ \theta_k^T|G_0^T \sim G_0^T & \theta_k^H|G_0^H \sim G_0^H \\ Z_n|\pi \sim \pi & \\ T_n|\theta_k^T \sim P^T(\theta_{Z_n}^T) & \\ U_n|Q_n, Z_n, \theta_k^H \sim P^H(U_n|Q_n, Z_n, \theta_k^H) & \\ L_n|T_n, \theta_s^L \sim P^L(L_n|T_n, \theta_{s(x_n)}^L) & \end{array} \tag{3}$$

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

$$\begin{array}{l} P(U_n, Q_n, Z_n, L_n, T_n, \pi_1, \pi_2, \ldots, \theta_1^T, \theta_2^T, \ldots, \theta_1^H, \theta_2^H, \ldots|\alpha, G_0^T, G_0^H, \theta_s^L) \\ = \prod_{i=1}^{\infty}\{P^L(L_n|T_n, \theta_{s(x_n)}^L)[P^T(T_n|Z_n, \theta_i^T)P^H(U_n|Q_n, Z_n, \theta_i^H)]^{\chi\{Z_n=i\}} \\ P(Z_n = i|\pi_1, \pi_2, \ldots)P(\theta_i^T|G_0^T)P(\theta_i^H|G_0^H)\}\prod_{i=1}^{\infty}P(\pi_i|\alpha, \pi_1, \ldots, \pi_{i-1}) \end{array} \tag{4}$$

where

$$\begin{array}{l} P^H(U_n|Q_n, Z_n, \theta_k^H) \\ = \prod_{m=1}^{M_n}[P(u_{n,m}|q_{n,m}, Z_n, \theta_k^H)P(q_{n,m}|q_{n,m-1}, Z_n, \theta_k^H)] \end{array} \tag{5}$$

and $\chi_{\{Z_n=i\}} = 1$ if $Z_n = i$ and 0 otherwise.

4. INFERENCE

As described above, Equation 5 provides the basic formulation of the graphical model. For simplicity, we let $\boldsymbol{O}$, $\boldsymbol{U}$, and $\boldsymbol{\varphi}$ be the set of observable variables, which include all L_n and U_n, where $1 \le n \le N$, the set of unobservable variables, which include all Z_n, T_n, θ_k^T, θ_k^H and π_k, where $1 \le n \le N$ and $1 \le k \le \infty$, and the set of model parameters, which include α, G_0^T, G_0^H, and θ_s^L respectively. Given a set of text fragment and the parameters $\boldsymbol{\varphi}$, the unsupervised learning problem can be viewed as an inference problem defined as follows:

$$\begin{array}{rcl} U^* & = & \arg\max_u\{P(\boldsymbol{U} = u|\boldsymbol{O}, \boldsymbol{\varphi})\} \\ & = & \arg\max_u\{\log P(\boldsymbol{U} = u|\boldsymbol{O}, \boldsymbol{\varphi})\} \end{array} \tag{6}$$

Since the computation of $\log P(\boldsymbol{U}|\boldsymbol{O}, \boldsymbol{\varphi}) = \log\frac{\int P(\boldsymbol{U},\boldsymbol{O}|\boldsymbol{\varphi})d\boldsymbol{O}}{P(\boldsymbol{O}|\boldsymbol{\varphi})}$ involves the marginalization of $P(\boldsymbol{U}, \boldsymbol{O}|\boldsymbol{\varphi})$, over the unobservable variables, exactly solving Equation 6 is intractable. As a result, approximation methods are required. We make use of Markov Chain Monte Carlo (MCMC) techniques to solve this problem in a principled and efficient manner.

In our graphical model, new density forms are available when the components Z_n, T_n, L_n, U_n and Q_n are used separately. Since conjugate priors are used in our model, we adopt Gibbs sampling to sample from the posterior distribution $P(\boldsymbol{U}|\boldsymbol{O}, \boldsymbol{\varphi})$. Based on the sampling process, we can determine how many distinct components are likely contributing to our data and what the parameters are for each component.

 # Unsupervised inference algorithm
INPUT: $\boldsymbol{X}$: The set of text fragments from different Web pages
OUTPUT: Z_n, T_n and U_n for all $x_n \in \mathcal{X}$
ALGORITHM:
0 set all model parameters as uninformative prior
1 until convergence
2 foreach $x_n \in \mathcal{X}$
3 sample Z_n according to Equations 7 and 8
4 update θ_k^T and θ_k^H for all k according to Equation 9
5 update T_n according to Equation 10
6 learn the HMM model corresponding to x_n and update θ_k^H using Baum-Welch algorithm
7 use the learned HMM model to label the text fragments using Viterbi algorithm
8 end foreach
9 end until

Figure 5: A high-level outline of our unsupervised inference algorithm.

Figure 5 depicts the high-level outline of our inference algorithm. We sample for the component indicator Z_n for the n-th text fragment as well as the component parameters θ_k^T and θ_k^H, for all $1 \le k \le \infty$. Assuming current state of the Markov chain in MCMC algorithm consists of $Z_1, Z_2, \ldots$, and the component parameters θ_k^T and θ_k^H, for all $1 \le k \le \infty$. For convenience, we use a variable W_n to represent the set of tokens $U_{n,1}, U_{n,2}, \ldots, U_{n,M_n}$ and labels $Q_{n,1}, Q_{n,2}, \ldots, Q_{n,M_n}$ of the n-th text fragment. Samples can be generated by repeating the following steps:

1. For $i = 1, \ldots, N$:

   - If Z_i is currently a singleton, remove $\theta_{Z_i}^T$ and $\theta_{Z_i}^H$ from the state.
   - Draw a new value for Z_i from the conditional distribution:

$$\begin{array}{l} P(Z_i = z|Z_{-i}, T_i, W_i, \theta_{Z_i}^T, \theta_{Z_i}^H) \\ = \begin{cases} \frac{N_{-i,c}}{N-1+\alpha}F(\theta_i^T, T_i)F(\theta_i^H, W_i), & \text{for existing } z, \\ \frac{\alpha}{N-1+\alpha}\int F(\theta^T, T_i)dG_0^T F(\theta^H, W_i)dG_0^H, & \text{for a new } z. \end{cases} \end{array} \tag{7}$$

   - If the new Z_i is not associated with any other observation, draw a value for $\theta_{Z_i}^T$ and $\theta_{Z_i}^H$ from:

$$\begin{array}{rcl} P(\theta^T|T_i) & \propto & F(\theta_i^T, T_i)G_0^T(\theta^T) \\ P(\theta^H|W_i) & \propto & F(\theta_i^H, W_i)G_0^H(\theta^H) \end{array} \tag{8}$$

2. For all $1 \le k \le \infty$:

   - Draw a new value for θ_k^T and θ_k^H from the posterior distribution based on the prior G_0^T and G_0^H and all the

data points currently associated with component k:

$$\begin{aligned} P(\theta^T|T_k) &\propto \prod_{i:Z_i=k} P(T_i|\theta^T)P(\theta^T) \\ &= \prod_{i:Z_i=k} F(\theta^T, T_i)G_0^T(\theta^T) \\ P(\theta^H|W_k) &\propto \prod_{i:Z_i=k} P(W_i|\theta^H)P(\theta^H) \\ &= \prod_{i:Z_i=k} F(\theta^H, W_i)G_0^H(\theta^H) \end{aligned} \quad (9)$$

As mentioned in Section 3, T_i can be modeled as a Bernoulli trial, let G_0^T be a Beta distribution, which is the conjugate prior of a binomial distribution, then $P(\cdot|\theta_k^T)$ can be modeled as a binomial distribution with the parameter θ_k^T. So the posterior probability $P(\theta^T|T_k)$ is also a Beta distribution. Similarly, let G_0^H be a Dirichlet distribution, which is the conjugate prior of a mixture model then $P(\cdot|\theta^H)$ can be modeled as a multinomial distribution with parameter θ^H and its posterior probability $P(\theta^H|W_k)$ is a Dirichlet distribution.

Our framework can consider the page-dependent layout format of text fragments to improve extraction. Considering the fact that Web pages within one Web site usually share the same set of layout information, we use a set of parameters θ_s^L to represent the layout information. Therefore, $P(\cdot|\theta_s^L, T_n)$ can be modeled as a multinomial distribution. Given θ_s^L, we can update T_n based on the $P(T_n|\theta_s^L, T_n)$.

$$P(T_n|L_n) \propto P(L_n|\theta_s^L, T_n)P(T_n)P(\theta_s^L) \quad (10)$$

After updating θ^H for all the components, the n-th text fragment will be labeled by the corresponding HMM generated from the k-th component, where $Z_n = k$. θ_k^H contains a set of HMM parameters: the start probability representing at which label the HMM starts, the transition probability representing the change of labels in the underlying Markov chain, and the emission probability representing which token would be generated by each label. We conduct Baum-Welch algorithm to derive the maximum likelihood estimate and update the set of probabilities. And the text fragment, also known as a token sequence, is labeled using Viterbi algorithm based on the updated parameters of the model.

To initialize this algorithm, we need to provide the parameters α, G_0^T , G_0^H, and θ_s^L. For the model parameters, α is the scaling parameter in the Dirichlet process, which essentially affects the number of normalized attributes in the normalization process. Since we apply our framework to the domains, for example, digital cameras, in which each product contains a number of attributes, we set α to a value that favors a large number of extracted attributes. G_0^T refers to the prior knowledge about how likely a text fragment will be a product attribute. We treat it as an uninformative prior by letting $\alpha = 1, \beta = 1$ of a Beta distribution. Similarly, G_0^H is treated as uninformative and all α's of a Dirichlet distribution are set to 1. θ_s^L can also be initialized in this way.

5. EXPERIMENTAL RESULTS FOR ATTRIBUTE NORMALIZATION

We have conducted several sets of experiments to evaluate our framework. The dataset used in our experiments is composed of data from three different domains, namely, digital camera, MP3 player, and LCD TV domains. For these domains, a set of Web pages were collected from different Web sites, which were randomly selected by making use of product search engines. Each Web page was first pre-processed to generate a set of text fragments as follows: A Web page is an HTML document mixed with ungrammatical text fragments and HTML tags. Each Web page can be represented by a (DOM) structure. A DOM structure is an ordered tree representing the layout format of a Web page. There are two kinds of nodes in a DOM structure. The first kind of nodes are the HTML nodes corresponding to the layout format of the Web page. These nodes are labeled with the corresponding HTML tags. The second kind of nodes are the text nodes, which are responsible for the text displayed in browsers. These nodes are simply labeled with the associated texts. We define a text fragment as the text within a block of information such as a line, a paragraph, a row of table, etc., conveying a single idea or message. To identify the text fragments, we select some HTML tags such as TR, BR, P, etc. We call these HTML tags and the corresponding HTML nodes in the DOM structure separators and separator nodes respectively. Consider a separator node, namely, $node_{seq}$, in a DOM structure. The texts contained in the text nodes that are offspring of $node_{seq}$ but do not have other separator nodes between $node_{seq}$ and the underlying text nodes are concatenated to form a text fragment.

Domain	No. of pages	No. of sites	No. of text fragments	No. of text fragments related to attributes
DC	50	21	5696	690
MP3	59	13	5040	572
LCD	61	15	3014	270

Table 2: A summary of the data used in the experiments collected from the digital camera, MP3 player, and LCD TV domains. DC, MP3, and LCD refer to the digital camera, MP3 player, and LCD TV domains respectively.

Human accessors were invited to prepare the ground truth of the data for evaluation. The reference attribute of the text fragment will be identified manually. If there was a conflict between the accessors, it was resolved by discussion among them. Note that such annotation is only used for evaluation purpose. Table 2 summarizes the information of the dataset. The first and second column of the table shows the total number of Web pages and the total number of Web sites from which the data is collected. The third column shows the total number of text fragments in all the Web pages after pre-processing. The fourth column shows the total number of text fragments about product attributes in all the pages.

In each domain, we conducted two sets of experiments. In the first set of experiments, we applied our framework to normalize the product attributes from all the Web pages in the domain. We call this set of experiments "Our Approach". The second set of experiment employs latent Dirichlet allocation (LDA) to cluster text fragments [2]. We call this set of experiments "LDA Approach". In this approach, the latent topics of LDA refer to different reference attributes. The probability, denoted by $P(T|\theta)$, that a token T is generated by the topic θ is first computed using LDA. Next, the reference attribute of a text fragment is determined by com-

	Digital camera			MP3 player			LCD TV		
	P	R	F	P	R	F	P	R	F
1	0.81	0.83	0.82	0.77	0.69	0.73	0.57	0.63	0.60
2	0.82	0.67	0.74	0.69	0.56	0.62	0.61	0.58	0.59
3	0.74	0.77	0.75	0.71	0.66	0.68	0.58	0.62	0.60
4	0.72	0.80	0.76	0.71	0.69	0.70	0.55	0.64	0.59
5	0.83	0.75	0.79	0.67	0.75	0.71	0.62	0.63	0.62
6	0.76	0.79	0.77	0.72	0.65	0.68	0.60	0.56	0.58
7	0.71	0.71	0.71	0.69	0.73	0.71	0.62	0.57	0.59
8	0.71	0.73	0.72	0.70	0.66	0.68	0.55	0.58	0.56
9	0.72	0.70	0.71	0.69	0.61	0.65	0.57	0.61	0.59
10	0.74	0.79	0.76	0.66	0.73	0.69	0.58	0.55	0.56
	0.76	0.75	0.75	0.70	0.67	0.69	0.59	0.60	0.59

Table 3: The attribute normalization performance of "Our Approach" on the digital camera, MP3 player, and LCD TV domains. Rows labeled from 1 to 10 refer to the performance of ten most common reference attributes in the domain. The last row refers to average performance in the domain. P, R, and F refer to the pairwise recall, precision, and F_1-measure respectively.

	Digital camera			MP3 player			LCD TV		
	P	R	F	P	R	F	P	R	F
1	0.63	0.18	0.28	0.56	0.23	0.33	0.64	0.19	0.29
2	0.56	0.17	0.26	0.55	0.22	0.31	0.54	0.18	0.27
3	0.66	0.23	0.34	0.61	0.20	0.30	0.59	0.20	0.30
4	0.55	0.23	0.32	0.51	0.25	0.34	0.49	0.15	0.23
5	0.45	0.20	0.28	0.54	0.22	0.31	0.51	0.16	0.24
6	0.40	0.26	0.32	0.58	0.27	0.37	0.55	0.19	0.28
7	0.40	0.29	0.34	0.53	0.28	0.37	0.42	0.13	0.20
8	0.39	0.27	0.32	0.60	0.24	0.34	0.55	0.20	0.29
9	0.42	0.29	0.34	0.57	0.25	0.35	0.53	0.16	0.25
10	0.36	0.27	0.31	0.53	0.20	0.29	0.55	0.18	0.27
	0.48	0.24	0.31	0.56	0.24	0.33	0.54	0.17	0.26

Table 4: The attribute normalization performance of "LDA Approach" on the digital camera, MP3 player, and LCD TV domains. Rows labeled from 1 to 10 refer to the performance of ten most common reference attributes in the domain. The last row refers to average performance in the domain. P, R, and F refer to the pairwise recall, precision, and F_1-measure respectively.

puting the highest joint probability that the tokens within the text fragment are generated by the same topic. Note that the tokens are independent and the order of the tokens is ignored in the LDA approach. The number of reference attributes, denoted by K, has to be fixed in advance, while our approach can handle unlimited number of product attributes.

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, 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.

Tables 3 and 4 show the performance of "Our Approach" and "LDA Approach" in three domains respectively. In each domain, we show the normalization performance of the ten most common reference attributes and the average normalization performance. For example, in the digital camera domain, the ten most common reference attributes include *dimension*, *battery*, etc. It can be observed that the performance of "Our Approach" is better than that of "LDA Approach". In particular, the average F_1-measure of "Our Approach" are 0.75, 0.69, and 0.59 in the digital camera, MP3, and LCD TV domains respectively, while the overall performance of "LDA Approach" are 0.31, 0.33, and 0.26 in the digital camera, MP3, and LCD TV domains respectively. Table 5 shows the top 5 weighted terms in the 10 reference attributes in the digital camera domain identified in our framework. It can be observed that the semantic meaning of the attributes can be easily interpreted from the terms.

Att. 1	Att. 2	Att. 3	Att. 4	Att. 5
usb	scene	red	lcd	white
cable	mode	eye	color	flash
connection	selector	reduction	tft	balance
direct	shooting	flash	matrix	daylight
compatible	portrait	mode	pixels	cloudy
Att. 6	Att. 7	Att. 8	Att. 9	Att. 10
manufacturer	package	length	optical	battery
brand	content	dimension	zoom	charge
product	quick	width	lens	rechargeable
number	start	depth	optics	included
corporation	guide	height	resolution	information

Table 5: The visualization of the top five weighted terms in the ten normalized attributes in the digital camera domain.

6. AUTOMATIC DOMAIN ONTOLOGY DISCOVERY

Domain ontology discovery aims at constructing a hierarchical structure of the reference attributes of a particular product domain. Each node of the ontology refers to a concept of the domain, while the internal nodes refer to some abstract concepts. A concept in an ontology is represented by a set of terms. Figure 3 shows some parts of the ontology generated from our experiments in the digital camera domain. The leave nodes of the ontology include *battery*, *charger*, etc., which are regarded as some finer-grained concepts. The internal node labeled as *power info* is an abstract concept summarizing the leave nodes *battery* and *charger* because these two concepts are highly related.

We have developed an approach to accomplishing the task of domain ontology discovery based on hierarchical agglomerative clustering. The outline of our ontology discovery approach is depicted in Figure 6. As we discussed above, the reference attributes identified by our framework are associated with a set of terms with probabilities. We represent each reference attribute A_i as $\{< t_1, w^i_{t_1} >, < t_2, w^i_{t_2} > , \cdots\}$, where $w^i_{t_k}$ refers to the weight of term t_k in A_i. We set the weight of a term to the probability that this term is generated by the reference attribute. These reference attributes are treated as singleton cluster in our ontology discovery method. We define the similarity between two clusters, say

```
# Ontology discovery algorithm
INPUT: A: The set of reference attributes
       Ψ: Similarity threshold
OUTPUT:domain ontology O
ALGORITHM:
0  C ← all singletons from A
1  foreach C' in C
2     create a leaf concept in O s.t. the concept
        is represented by the top K weighted terms in C'
3  end foreach
4  until convergence
5     let C_i and C_j be the pair of clusters in C
          s.t. sim(C_i, C_j) is the largest
6     if sim(C_i, C_j) > Ψ
7        create a new cluster C^new
8        foreach t' in C^new
9           w_t'^new ← w_t'^i + w_t'^j
10       end foreach
11       normalize all the w_t'^new in C^new
12       C ← C \ {C_i, C_j} ∪ {C^new}
13       create an abstract concept as the parent of
             the concepts representing C_i and C_j
14    end if
15 end until
```

Figure 6: An outline of our ontology discovery algorithm.

C_i and C_j as the cosine similarity of the terms as follows:

$$sim(C_i, C_j) = \frac{\sum_{t'} w^i_{t'} w^j_{t'}}{\sqrt{(\sum_{t'} {w^i_{t'}}^2)(\sum_{t'} {w^j_{t'}}^2)}} \quad (11)$$

The two clusters with the highest similarity will be merged into a single cluster. The weights of the terms in the newly formed cluster are the sum of the weights of the terms in the original clusters, normalized by the total sum of the weights of all the terms in the new cluster. This process iterates until no more clusters can be formed, or all the similarity values between any pair of clusters are less than a certain threshold, namely, Ψ.

7. EXPERIMENTAL RESULTS FOR ONTOLOGY CONSTRUCTION

We have conducted experiments to evaluate the performance of our ontology discovery approach using the dataset summarized in Table 2. In each domain, human accessors were invited to manually construct the domain ontology after browsing all the Web pages collected in our experiments. If there was a conflict between the accessors, it was resolved by discussions among them. This manually constructed domain ontology is used as the ground truth in our experiments. In each domain, we apply our automatic ontology discovery approach to the reference attributes identified in our attribute normalization approach. The discovered ontology is compared with the manually constructed one to evaluate the performance.

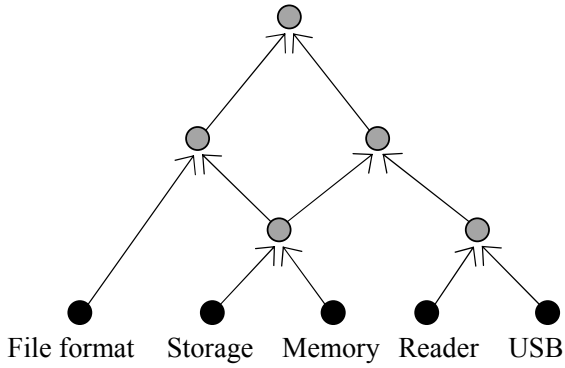


Figure 7: A part of the ontology in the MP3 player domain.

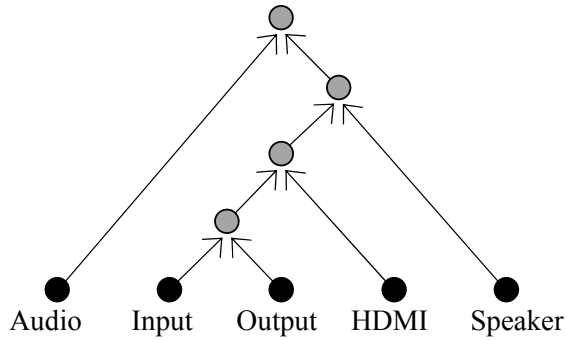


Figure 8: A part of the ontology in the LCD TV domain.

We aim at evaluating the effectiveness of discovering concepts of the domain ontology. We adopt the precision and recall as the evaluation metric. Recall is defined as the number of concepts that are correctly discovered by the system, divided by the actual number of concepts in the manually constructed ontology. Precision is defined as the number of concepts that are correctly predicted by the system, divided by the total number of concepts in the ontology constructed by the system. F_1-measure is defined as the harmonic mean of equal weighting of recall and precision.

The precision, recall, and F_1-measure are 0.76, 0.68, and 0.72 respectively in the digital camera domain; 0.75, 0.71, and 0.73 respectively in the MP3 player domain; 0.67, 0.57, and 0.62 respectively in the LCD TV domain. It shows that our ontology discovery approach can effectively identify the domain ontology concepts. Figures 3, 7, and 8 illustrate parts of the ontologies generated in our experiments in the digital camera domain, MP3 player domain, and LCD TV domain respectively. It can be observed that the generated ontology can effectively organize the concepts in a hierarchy. For example, the reference attribute *charger* and *battery* have the same parent, which can be interpreted as the concept about power information.

8. RELATED WORK

As mentioned in Section 1, Guo et al. proposed a method to categorize product features [8]. The major limitation is that they focus on Web opinions and require the opinions

contain the *Pros* and *Cons* columns for extracting product features. Product attribute normalization problem is related to the task of record resolution. Record resolution is the problem of determining which records in a database refer to the same entities, and is a crucial and expensive step in data organization. Singla and Domingos [13] developed an approach to record resolution based on Markov Logic Network. Their approach is to formulate first-order logic and probabilistic graphical models and combine them in Markov logic by attaching weights to first-order formulas, and viewing them as templates for features of Markov networks. Experiments on two citation databases showed that the resulting learning and inference problems can be solved efficiently. Bhattacharya and Getoor [1] proposed an unsupervised approach for record resolution based on Latent Dirichlet Allocation (LDA). A probabilistic generative model was developed for collectively resolving entities in relational data, which did not make pairwise decisions and introduced a group variable to capture relationships between entities. One limitation of these approaches is that the entities are required to be extracted in advance and cannot be applied to raw data.

Some methods have been developed to jointly extract information and conduct data mining, sharing similar goal to our framework. Wellner et al. described an approach to integrated inference for extraction and coreference based on conditionally-trained undirected graphical models [17]. They advocated conditional-probability training to allow free use of arbitrary non-independent features of the input, and adapted undirected graphical models to represent autocorrelation and arbitrary possibly cyclic dependencies. Also approximate inference and parameter estimation were performed in these large graphical models by structured approximations. However, the attributes to be extracted have to be known in advance in these approaches and previously unseen attributes cannot be handled.

Several supervised information extraction methods such as wrappers have been proposed to handle semi-structured documents like Web pages [5, 11, 9, 14, 21]. However, these supervised methods require human effort in preparing training examples, and the learned extraction rules can only extract those product attributes pre-specified in the training examples in advance. Several methods have been developed to extract data from Web pages without supervision [6, 3]. Liu et al. [10] proposed a system known as MDR (Mining Data Records in Web pages) to discover the data region in a Web page by making use of the repeated pattern in HTML tag trees. This system firstly builds a HTML tag tree of the page, then conducts mining algorithms in data regions in the page using the tag tree and string comparison, and heuristics are then applied to extract useful information from the data region. 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. [7] applied hidden Markov model and exploited prior knowledge to extract information in an unsupervised manner. They demonstrated that for certain field structured extraction tasks, such as classified advertisements and bibliographic citations, small amounts of prior knowledge could be used to learn effective models in a primarily unsupervised fashion, which could dramatically improve the quality of the learned structure. However, the quality of the extracted data was unlikely suitable for subsequent data mining tasks.

9. CONCLUSIONS AND FUTURE WORK

We have developed a framework for normalizing Web product attributes from Web pages without the need of manually labeled training examples. The Web pages can be collected from different Web sites and hence contain different layout format and content. We have designed a probabilistic graphical model for generating text fragments in Web pages. The model is incorporated with Hidden Markov Models (HMM) considering attribute names and attribute values of text fragments. Attribute normalization is accomplished by conducting inference over the graphical model. Dirichlet process is employed to handle the unlimited number of product attributes in a domain. An unsupervised inference method is proposed to infer the reference attribute to which a text fragment belongs. We have conducted extensive experiments using real-world data of three different domains to show the effectiveness of our framework.

We have also developed an application for automatically discover the domain ontology by making use of the reference attributes obtained in attribute normalization. The discovered ontology can effectively organize the concepts in a hierarchical structure. Experiments have been conducted to demonstrate the efficacy of our approach.

One possible direction is to make use of the automatically constructed domain ontology for other intelligent tasks such as shopping agents. The constructed ontology can effectively represent the knowledge of the Web products. Hence, complex query or inference can be supported to assist customers to make decision.

10. 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 (SDM)*, pages 47–58, 2006.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993 – 1022, 2003.

[3] 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.

[4] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan. A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428, 2006.

[5] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan. A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428, 2006.

[6] P. Golgher and A. da Silva. Bootstrapping for example-based data extraction. In *Proceedings of the Tenth ACM International Conference on Information and Knowledge Management (CIKM)*, pages 371–378, 2001.

[7] T. Grenager, D. Klein, and C. Manning. Unsupervised learning of field segmentation models for information extraction. In *Proceedings of the Forty-Third Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 371–378, 2005.

[8] H. Guo, H. Zhu, Z. Guo, X. Zhang, and Z. Su.

Product feature categorization with multilevel latent semantic association. In *Proceeding of the Eighteenth ACM conference on Information and Knowledge Management (CIKM)*, pages 1087–1096, 2009.

[9] 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.

[10] 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.

[11] X. Meng, H. Wang, D. Hu, and C. Li. A supervised visual wrapper generator for web-data extraction. In *Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC)*, pages 657–662, 2003.

[12] S. Ravi and M. Paşca. Using structured text for large-scale attribute extraction. In *Proceeding of the Seventeenth ACM conference on Information and Knowledge Management (CIKM)*, pages 1183–1192, 2008.

[13] P. Singla and P. Domingos. Entity resolution with markov logic. In *Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM)*, pages 572–582, 2006.

[14] C. Sutton, K. Rohanimanesh, and A. McCallum. Dynamic conditional random fileds: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of Twenty-First International Conference on Machine Learning (ICML)*, pages 783–790, 2007.

[15] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.

[16] J. Turmo, A. Ageno, and N. Catala. Adaptive information extraction. *ACM Computing Surveys*, 38(2), 2006.

[17] 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.

[18] T.-L. Wong, W. Lam, and T. Wong. An unsupervised framework for extracting and normalizing product attributes from multiple web sites. In *Proceedings of the Thirty-First Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 35–42, 2008.

[19] B. Wu, X. Cheng, Y. Wang, Y. Guo, and L. Song. Simultaneous product attribute name and value extraction from web pages. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 295–298, 2009.

[20] N. Yoshinaga and K. Torisawa. Open-domain attribute-value acquisition from semi-structured texts. In *Proceedings of the sixth International Semantic Web Conference (ISWC-07)*, pages 55–66, 2007.

[21] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and H.-W. Hon. Webpage understanding: an integrated approach. In *Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 903–912, 2007.