An Unsupervised Approach for Product Record Normalization across Different Web Sites*

Tak-Lam Wong[1] and **Tik-Shun Wong**[2] and **Wai Lam**[2]
[1]Department of Computer Science and Engineering
The Chinese University of Hong Kong, Hong Kong
wongtl@cse.cuhk.edu.hk
[2]Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong, Hong Kong
{tswong,wlam}@se.cuhk.edu.hk

Abstract

An unsupervised probabilistic learning framework for normalizing product records across different retailer Web sites is presented. Our framework decomposes the problem into two tasks to achieve the goal. The first task aims at extracting attribute values of products from different sites and normalizing them into appropriate reference attributes. This task is challenging because the set of reference attributes is unknown in advance. Besides, the layout formats are different in different Web sites. The second task is to conduct product record normalization aiming at identifying product records referring to the same reference product based on the results of the first task. We develop a graphical model for the generation of text fragments in Web pages to accomplish the two tasks. One characteristic of our model is that the product attributes to be extracted are not required to be specified in advance and an unlimited number of previously unseen product attributes can be handled. We compare our framework with existing methods. Extensive experiments using over 300 Web pages from over 150 real-world Web sites from three different domains have been conducted demonstrating the effectiveness of our framework.

Introduction

The readily accessible Internet provides a convenient and cost-saving environment for both retailers and consumers. Many retailers have set up Web sites containing catalogs of products. Consumers can shop around over the Internet by browsing retailer Web sites. Recently, several specialized search engines have been developed for users to search and compare products from different retailer Web sites[1]. Such systems can help users match the same product from different retailer sites and find the best deal. One limitation of such systems is that retailers are required to manually input the value for each attribute of products to the database of the search engine via an interface. This may lead to out-of-date information of products resulting in degradation of user satisfaction. Moreover, the attributes of products in these systems are just simple and general fields pre-defined by the search engines, for example, brand name, model number, brief description, and price. However, there may exist some domain specific attributes such as the attribute "resolution" in the digital camera domain. These domain specific attributes are important because they can help users retrieve, analyze, and compare the products.

Figure 1: A sample of a portion of a Web page showing a digital camera collected from a retailer Web site. (Web site URL: *http://www.superwarehouse.com*)

Another problem is that it is not easy to determine whether product records from different Web sites refer to the same product. For instance, some Web sites may use different model numbers, or no model number for the same digital camera. Examples of such situations can also be observed in the data set used in our experiments. Figures 1 and 2 depict two Web pages collected from two different retailer Web sites. Both pages consist of the same product, but with different product names. To resolve the products, human effort and expert knowledge are required.

Product record normalization is defined as the clustering of the same/similar product records into the same group. Bilenko et al. proposed a product normalization approach which aims at computing the similarity between products stored in a structured database (Bilenko, Basu, and Sahami 2005). Their method considers the linear combination of different basic similarity functions related to each field of records. One limitation of this approach is that it requires structured records with a fixed set of attributes. As a result, attribute values of each product need to be extracted manually in advance. Alternatively, attribute values may be extracted from Web sites automatically by making use of wrappers (Rurmo, Ageno, and Catala 2006; Chang et al. 2006; Zhao, Meng, and Yu 2007; Sarawagi and Cohen 2004; Viola and Narasimhan 2005; Wong and Lam 2007). How-

*The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Nos: CUHK4193/04E and CUHK4128/07) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050363 and 2050391). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

[1]Examples are Google Product Search (http://www.google.com/products/) and Shopping.com (http://www.shopping.com).

Figure 2: A sample of a portion of a Web page showing the same digital camera to the one depicted in Figure 1, but collected from a different retailer Web site. (Web site URL: *http://www.crayeon3.com*)

ever, a learned wrapper for a Web site cannot be applied to other site for information extraction because the layout formats are different. Consequently, each Web site needs to learn its own wrapper and training examples are required for every site. As a result, this approach will be infeasible for handling numerous retailer Web sites. Another limitation of the approach proposed by Bilenko et al. is that human effort is needed to prepare training examples of product normalization for learning the weights in the linear combination. Product normalization shares certain resemblance with the research area of duplicate detection or record linkage in database (Bilenko and Mooney 2003; Sarawagi and Bhamidipaty 2002; Ravikumar and Cohen 2004; Culotta et al. 2007). However, these approaches aim at matching records which have a fixed set of attributes in database. Therefore, they are not applicable to our problem in which attributes of products can be previously unseen and the number of attributes is unknown.

In this paper, we aim at developing an unsupervised framework which can automatically conduct product record normalization across different retailer Web sites. To achieve this, our framework can also extract and normalize the domain specific attribute values of products. This can help users analyze the products. It is particularly useful when there is no identifier for products. We develop a probabilistic graphical model which can model the generation of text fragments in Web pages. Based on this model, our framework decomposes product record normalization into two tasks. The first task is the product attribute values extraction and normalization task. This task aims at automatically extracting text fragments related to some domain specific attributes from Web pages and clustering them into appropriate reference attributes. One characteristic of our approach is that it can handle Web pages with different layout formats. Another characteristic is that it can handle previously unseen attributes and an unlimited number of attributes. The second task is the product record normalization task. We tackle this task by considering the similarity between products based on the results from the first task. Product record normalization is then accomplished by another level of unsupervised learning. We have conducted extensive experiments using over 150 real-world retailer Web sites from three different domains. We have compared our framework with existing methods and the results demonstrate the effectiveness of our approach.

Problem Definition

Consider a collection of reference products $\mathcal{P}$ in a domain $\mathcal{D}$. Each product $p_i \in \mathcal{P}$ is characterized by the values of a set of reference attributes $\mathcal{A}$. For example, in the digital camera domain, reference attribute may include "resolution", "sensor type", etc. The product shown in Figure 1 has a value of "10 Megapixel" for the reference attribute "resolution". We let v_i^p be the value of the attribute $a_i \in \mathcal{A}$ for the reference product p. Notice that $\mathcal{A}$ is domain specific and the number of elements in $\mathcal{A}$ is unknown. Suppose we have a collection of product records $\mathcal{R}$ which refers to the set of realization of some products $p \in \mathcal{P}$. For example, Figures 1 and 2 show two different product records. We let r_i be the i-th product record in $\mathcal{R}$ and $r_i.U = p$ if r_i is a realization of the reference product p. Notice that each reference product p may have several product records, while a product record r can only be a realization of a particular reference product. For example, the product records in Figures 1 and 2 refer to the same reference product. For each product record r, we let $v_i^p(r)$ be the realization of the value of the reference attribute a_i of the product p for the product record r. For example, the attribute values of the reference attribute "light sensitivity" are "Auto, High ISO, ISO 80/100/200/400/8000/1600, equivalent" and "Auto ISO 80/100/200/400/800/1600" in Figures 1 and 2 respectively.

We consider a collection of Web pages $\mathcal{C}$ which are collected from a collection of Web sites $\mathcal{S}$. Each Web page $c \in \mathcal{C}$ contains a product record r. The Web page c can be considered as a set of text fragments $\mathcal{X}$. For example, "Features" and "10 Megapixel" are samples of text fragments in the Web page shown in Figure 1. Each text fragment $x \in \mathcal{X}$ may refer to an attribute value $v_i^p(r)$. We let $x.T = 1$ if x refers to an attribute value of a product record, and 0 otherwise. Moreover, we have $x.A = a_i$ if x refers to the reference attribute $a_i \in \mathcal{A}$. We also define $x.C$ and $x.L$ as the content and the layout format of the text fragment x respectively. For example, the content of the text fragment "10 Megapixel" in Figure 1 can be the terms contained. The layout format of the same text fragment can be the color, font size, etc. Notice that $x.T$ and $x.A$ are unobservable, whereas $x.C$ and $x.L$ are observable. As a result, we can define product record normalization as follows:

Product record normalization: Given two product records r_{c_i} and r_{c_j} contained in Web page c_i and c_j, product normalization aims at predicting whether $r_{c_i}.U = r_{c_j}.U$.

To support this, the attribute values of reference attributes for a product record are required to be determined in advance. Therefore, we define attribute extraction and normalization as follows:

Attribute extraction: Given a collection of Web pages $\mathcal{C}$. Each page $c \in \mathcal{C}$ contains a record $r \in \mathcal{R}$. The goal of attribute extraction is to discover all text fragments $x \in \mathcal{X}$ such that $x.T = 1$, given $x.C$ and $x.L$.

Attribute normalization: Given a collection of text fragments such that $x.T = 1$ for all text fragments in the collec-

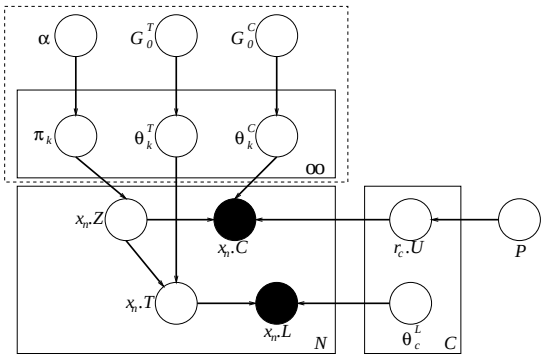

Figure 3: The graphical model for the generation of text fragments in Web pages. Shaded nodes and unshaded nodes represent the observable and unobservable variables respectively. The edges represent the dependence between variables and the plates represent the repetition of variables.

tion, the goal of attribute normalization is to cluster the text fragments x_i and x_j into the same group if $x_i.A = x_j.A$, given $x.C$ and $x.L$.

Formally, we consider the probabilities $P(x.T|x.C, x.L)$ and $P(x.A|x.C, x.L)$ in attribute extraction and attribute normalization respectively. However, tackling these two problems separately may lead to conflict solution because actually $P(x.T|x.C, x.L)$ and $P(x.A|x.C, x.L)$ are dependent. Hence, in our framework, we consider the joint probability $P(x.T, x.A|x.C, x.L)$ in a single framework facilitating cooperative interaction among both tasks. The output of attribute extraction and attribute normalization can support the task of product normalization. In our framework, we design another unsupervised learning method which can effectively solve product normalization based on the results from the attribute extraction and normalization task.

Our Model

We develop a graphical model, which can model the generation of text fragments in a Web page, for solving the product record normalization task. To achieve this, our model also considers the attribute extraction and normalization task. We employ Dirichlet process prior in our model leading to the characteristic that it can handle unlimited number of reference attributes. In essence, our model can be considered as a Dirichlet mixture model. Each mixture component refers to a reference attribute. Figure 3 shows the graphical model. We adopt the stick breaking construction representation of Dirichlet process in our presentation (Blei and Jordan 2006; Teh et al. 2006). The block in dashed line refers to the stick breaking construction of Dirichlet process and interested readers can refer to (Blei and Jordan 2006; Teh et al. 2006).

Suppose we have a collection of Web pages collected denoted as $\mathcal{C}$. Each page $c \in \mathcal{C}$ contains a product record $r \in \mathcal{R}$. There is an unobservable variable $r_c.U$ for each page c, depending on the variable P, which refers to the reference product in the domain. Therefore $r_{c_1}.U = r_{c_2}.U$ if the product records collected from pages c_1 and c_2 correspond to the same product. Suppose we have a collection of N different text fragments collected from $\mathcal{C}$ different Web pages. Let $x_n(c)$ represent the n-th text fragment collected from page c. Each text fragment consists of variables $x_n.T$, $x_n.C$, and $x_n.L$ as described in the previous section. There is another variable $x_n.Z$ which is to replace $x_n.A$. Essentially $x_n.Z = i$ if $x_n.A = a_i$ where $a_i \in \mathcal{A}$ showing the index of the reference attribute to which x_n corresponds. Each mixture component in our framework consists of its own distribution about the content and layout format of the text fragments referring to the attribute. θ_k^C is the parameter of the content distribution. G_0^C is the base distribution for generating θ_k^C in the Dirichlet process. Together with $r_c.U$ and $x_n.Z$, θ_k^C generates the content of a text fragment. For example, we model the content of text fragments by a mixture model of terms, θ_k^C is then the set of multinomial distribution parameters and G_0^C is a Dirichlet distribution using conjugacy. θ_k^L is the parameter for the distribution of $x_n.T$. θ_k^L is generated by another base distribute G_0^L. Since $x_n.T$ is binary, we treat θ_k^L and G_0^L to be binomial and beta distributions respectively. Together with $x_n.T$, θ_c^L, which refers to layout distribution parameter, will generate the layout format $x_n.L$. For example, a particular layout format can be considered as a Bernoulli trial depending on the parameter θ_c^L and $x_n.T$. The joint probability for generating a particular text fragment x_n collected from page c given the parameters α, G_0^C, G_0^T, p, θ_s^L can then be expressed as follows:

$$\begin{aligned} & P(x_n.C, x_n.L, x_n.T, x_n.Z, r_c.U, \pi_1, \ldots, \theta_1^C, \ldots, \theta_1^T, \ldots | \alpha, G_0^C, G_0^T, P, \theta_c^L) \\ = & \prod_{i=1}^{\infty} \{P(x_n.L|x_n.T, \theta_{c(x_n)}^T) P(x_n.Z = i|\pi_1, \pi_2, \ldots) \\ & P(\theta_i^C|G_0^C) P(\theta_i^T|G_0^T)\} \\ & \left[P(x_n.C|x_n.Z, \theta_i^C) P(x_n.T|x_n.Z, \theta_i^T)\right]^{\chi_{\{x_n.Z=i\}}} \\ & P(r_c.U|P) \prod_{i=1}^{\infty} P(\pi_i|\alpha, \pi_1, \ldots, \pi_{i-1}) \end{aligned} \tag{1}$$

where $\chi_{\{x_n.Z=i\}} = 1$ if $x_n.Z = i$ and 0 otherwise.

Product record normalization aims at computing $P(r_c.U|x_n.C, x_n.L; \varphi)$. However, the computation is intractable since it involves the marginalization of unobservable variables. We tackle this problem by decomposing the problem into two tasks. The first task is the attribute extraction and normalization task. In this task, we consider that $x_n.C$ only depends on $x_n.Z$ as well as θ_k^C and compute $P(x_n.T, x_n.Z, |x_n.C, x_n.L; \varphi)$. The second task makes use of the results from the first task. We derive an unsupervised method to predict whether $r_{c_1}.U = r_{c_2}.U$ based on the extracted and normalized attributes from pages c_1 and c_2.

Attribute Extraction and Normalization

In attribute extraction and normalization, we consider that $x_n.C$ only depends on $x_n.Z$ and θ_k^C. We employ the truncated stick breaking process (Ishwaran and James 2001) and variational method to tackle this problem. Denote $\mathcal{O}$ and $\mathcal{U}$ for the observable and unobservable variables. The idea of variational method is to design a distribution $Q(\mathcal{U}|\nu)$ where ν is called the set of variational parameters to approximate $P(\mathcal{U}|\mathcal{O}, \varphi)$. It can be shown that it is equivalent to maximizing $E_Q[\log P(\mathcal{O}, \mathcal{U}|\varphi)] - E_Q[\log Q(\mathcal{U}|\nu)]$ where $E_Q[Y]$ is the expected value of Y over probability distribution Q. In particular, we define Q as follows:

$$\begin{aligned} & Q(\boldsymbol{x.T}, \boldsymbol{x.Z}, \boldsymbol{\theta^C}, \boldsymbol{\theta^T}, \boldsymbol{\pi}) \\ = & \prod_{k=1}^{K-1} Q_\pi(\pi_k|\tau_{k,0}, \tau_{k,1}) \prod_{k=1}^{K} Q_{\theta^T}(\theta_k^T|\delta_{k,0}, \delta_{k,1}) \\ & \prod_{k=1}^{K} Q_{\theta^C}(\theta_k^C|\zeta) \prod_{n=1}^{N} Q_T(x_n.T|\omega_n) \prod_{n=1}^{N} Q_Z(x_n.Z|\boldsymbol{\phi_n}) \end{aligned} \tag{2}$$

where K is the truncation level; $Q_\pi(\cdot|\tau_{k,0}, \tau_{k,1})$ is the Beta distribution with parameters $\tau_{k,0}$ and $\tau_{k,1}$; $Q_{\theta^T}(\theta_k^T|\delta_{k,0}, \delta_{k,1})$ is the Beta distribution with parameters $\delta_{k,0}$ and $\delta_{k,1}$; $Q_{\theta^C}(\theta_k^C|\boldsymbol{\zeta})$ is the Dirichlet distribution with parameter set $\boldsymbol{\zeta}$; $Q_T(x_n.T|\omega_n)$ is the binomial distribution with parameter ω_n; and $Q_Z(x_n.Z|\phi_{n,1}, \ldots, \phi_{n,K})$ is the multinomial distribution with parameter set $\phi_{n,1}, \ldots, \phi_{n,K}$. In the truncated stick breaking process, $Q_Z(x_n.Z|\boldsymbol{\phi_n}) = 0$ for $x_n.Z > K$. Under this setting, $E_Q[\log P(\mathcal{O}, \mathcal{U}|\varphi)] - E_Q[\log Q(\mathcal{U}|\nu)]$ can be expressed in closed form. Optimal value of each of the variation parameters can be determined by finding the first derivative and setting it to zero. The optimal values of the variational parameters are listed as follows:

$$\begin{array}{l} \tau_{k,0} = (1-\alpha) + \sum_{n=1}^{N} \phi_{n,k} \\ \tau_{k,1} = \alpha + \sum_{n=1}^{N} \sum_{j=k+1}^{K} \phi_{n,j} \end{array} \quad (3)$$

$$\begin{array}{l} \delta_{k,0} = \mu_0^T + \sum_{n=1}^{N} \omega_n \phi_{n,k} \\ \delta_{k,1} = \mu_1^T + \sum_{n=1}^{N} (1-\omega_n) \phi_{n,k} \end{array} \quad (4)$$

$$\zeta_{k,j} = \mu_j^C + \sum_{n=1}^{N} w_{n,j} \phi_{n,k} \quad (5)$$

$$\begin{array}{rcl} \phi_{n,k} & \propto & \exp\{\sum_{j=1}^{K-1} [\Psi(\tau_{j,1}) - \Psi(\tau_{j,0} + \tau_{j,1})] \\ & & +\Psi(\tau_{k,0}) - \Psi(\tau_{k,0} + \tau_{k,1}) \\ & & +\sum_{j=1}^{|V|} w_{n,j} [\Psi(\zeta_{k,j}) - \Psi(\sum_{h=1}^{|V|} \zeta_{k,h})] \\ & & +\omega_n (\Psi(\delta_{k,0}) - \Psi(\delta_{k,0} + \delta_{k,1})) \\ & & +(1-\omega_n)(\Psi(\delta_{k,1}) - \Psi(\delta_{k,0} + \delta_{k,1}))\} \end{array} \quad (6)$$

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.

$$\omega_n = \frac{1}{1+e^{-h(\phi_{n,k}, \delta_{k,0}, \delta_{k,1}, \theta_s^L)}} \quad (7)$$

where

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

and $u_{n,f} = 1$ if x_n contains the f-th layout format in F_c in page $c = c(x_n)$, and 0 otherwise. Given the model parameters, one can then apply the steepest ascent algorithm to find the values of the variational parameters. In particular, $\phi_{n,k}$ shows how likely x_n corresponds to $a_k \in \mathcal{A}$ and ω_n shows how likely x_n corresponds to an attribute value of a product record. Similarly, we can find the optimal value of $\theta_{c(x_n)}^L$ as follows:

$$\theta_{c,f}^L = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \omega_n u_{n,f} \quad (8)$$

Based on this idea, we have developed an unsupervised learning algorithm depicted in Figure 4 based on expectation maximization technique. This algorithm can leverage the content of text fragments, as well as the format layout of text fragments to conduct extraction and normalization. Our algorithm only requires a list of a few attribute related terms in the domain. As a result, compared with many existing methods, our proposed model significantly reduces human effort. Readers can refer to (Wong, Lam, and Wong 2008) for more detailed description of our algorithm.

 # Attribute extraction and normalization algorithm

INPUT: K: Truncation level of truncated stick breaking process
$\boldsymbol{X}$: The set of text fragments from different Web pages
$\boldsymbol{\kappa}$: A list of terms related to product attributes

OUTPUT: $\phi_{n,k}$ and ω_n for all $x_n \in \boldsymbol{X}$

INIT:

0 set all model parameters as uninformative prior
1 set all $\zeta_{k,j}$ to zero for all $1 \leq k \leq K, 1 \leq j \leq |V|$
2 set all $\delta_{k,0}$ to 4 and $\delta_{k,1}$ to 6 for all $1 \leq k \leq K$
3 for $i = 1, \ldots, |\boldsymbol{\kappa}|$
4 set $\zeta_{i,j}$ to a value ≥ 1 if $\kappa_i = v_j$
5 set $\delta_{i,0} > \delta_{i,1}$
6 end for
7 foreach $x_n \in \boldsymbol{X}$
8 Compute $\phi_{n,k}$ according to Equation 6
9 Compute ω_n according to Equation 7
10 end foreach
11 until convergence

E-Step

12 Invoke steepest ascent algorithm according to Equations 3-7

M-Step

13 Compute all $\theta_{c,f}^L$ according to Equation 8

Figure 4: An outline of our unsupervised inference algorithm.

Product Record Normalization

Product normalization aims at determining how likely that $r_{c_1}.U = r_{c_2}.U$ based on the extracted and normalized attributes from pages c_1 and c_2. For example, "Auto, High ISO Auto, ISO 80/100/200/400/800/1600 equivalent" and "Auto ISO 80/100/200/400/800/1600" are extracted from Figures 1 and 2 respectively and normalized to the same reference attribute. In our approach, we first calculate the similarity for each reference attribute between product records. The similarity between product records is then computed based on the individual reference attribute similarities.

For a particular attribute $a_i \in \mathcal{A}$ discovered, the attribute value $v_i(r)$ of a product record r basically contains a set of tokens. As a result, we can measure the similarity of attribute a_i between two different product records based on the token content. For each attribute a_i, we first compute the term frequency-inverse document frequency (TF-IDF) weight for each distinct token in the attribute values $v_i(r)$. In our model, TF is defined as the number of occurrence of the token divided by the total number of tokens in the attribute value for a_i. IDF is defined as the logarithm of the total number of product records divided by the number of records containing the token in $v_i(r)$. Next, we employ the cosine similarity, denoted as $Sim_{a_i}(r_{c_1}, r_{c_2})$, to represent the similarity in attribute a_i between the two products collected from pages c_1 and c_2

After computing the similarities for individual attributes, the overall similarity between products can be computed. We observe that common attributes such as "resolution", "sensor" in the digital camera domain are good indicators for product record normalization because they are likely to be mentioned in different Web sites and can be used for measuring the similarities. However, some rare attributes which are seldom mentioned may not be useful in this task. Therefore, we define the weight w_{a_i} of attribute a_i as follows:

$$w_{a_i} = N_{a_i}(r)/N(r) \quad (9)$$

where $N(r)$ and $N_{a_i}(r)$ denote the total number of records and the number of records containing attribute a_i respectively. The overall similarity between product records can be defined as the linear combination of the similarities in attributes between products as follows:

$$Sim(r_{c_1}, r_{c_2}) = \sum_{a' \in \mathcal{A}} w_{a'} Sim_{a'}(r_{c_1}, r_{c_2}) \quad (10)$$

Attr.	Attribute Values
A#1	Supported Battery Details 1 x Li - ion rechargeable battery (included)
	Supported Battery: Sony NP-FH40
	Battery Charger AC - L200
	InfoLITHIUM Battery with AccuPower Meter System (NP-FH60)
	rechargeable lithium - ion battery
A#2	16:9 Wide Screen Mode
	LCD Monitor: 2.7 - inch
	2.7" LCD Monitor
	0.35", 114,000 pixels, wide-screen
	LCD Size 2.7 inches
	Display Screen 2 " Active Matrix TFT Color LCD 200 Kilopixels

Table 1: Samples of attribute values extracted and normalized from different Web sites in the camcorder domain.

Based on the similarities between products, product record normalization is then accomplished by invoking the hierarchical agglomerative clustering method similar to the one described in (Bilenko, Basu, and Sahami 2005), in which similar product records will be identified and form a cluster. In the algorithm, the canopy method is applied to reduce the complexity (McCallum, Nigam, and Ungar 2000).

Experimental Results

We have conducted several sets of experiments using over 150 real-world retailer Web sites in three different domains, namely, camcorder, MP3 player, and digital camera, to evaluate the performance of our framework in conducting product normalization. In particular, there are 96 Web pages from 62 sites, 111 Web pages from 61 sites, and 85 Web pages from 41 sites, in the camcorder, MP3 player, and digital camera domains respectively. Each page contains a product record referring to a reference product. Notice that the layout formats of Web pages are greatly different and it is infeasible to use information extraction wrappers to extract product attribute values. To prepare the gold standard for evaluation purpose, each product record is labeled by human experts with the reference product to which the product record corresponds.

We apply our framework to the collected Web pages as described above for product normalization. We compare our framework with the state-of-the-art existing work for product normalization proposed by Bilenko and Mooney (Bilenko, Basu, and Sahami 2005). We call this method BM approach in this paper. Their approach requires pre-processed dataset that includes several fields such as "product name", "brand", "price", and "raw description" of each product record. We manually extracted these attributes for each record. Since some pages do not explicitly mention the attributes such as "brand", we manually determined the values for the record in such cases. Such information is only provided for the BM approach. As their approach requires learning the weight in calculating the similarity between products, manually labeled training examples are also needed. As a result, the data pre-processing and the preparation of training examples require extensive human work and expertise. In contrary, our framework is carried out in a fully unsupervised manner and requires very little human effort.

In each domain, the experimental data are randomly divided into two disjoint sets, in which there is no overlapping of products. In each run of the experiment, one set is used as training examples, while the other set is used as testing data. This is to simulate the real-world situation that products are probably previously unseen. For the BM approach, we first learn a model using the training examples. The learned model is then applied to the testing data for product normalization. Since our framework is unsupervised and does not require training, we can directly apply our framework to the testing data and compare the results with the one obtained using BM approach. We adopt recall and precision as the evaluation metrics in the experiments. Recall is defined as the number of pairs of product records which are correctly identified as the same product divided by the total number of record pairs in the data. Precision is defined as the number of pairs of product records which are correctly identified as the same products divided by the total number of record pairs identified by the system.

Figures 5 depicts the recall-precision plot for the results of product record normalization in the camcorder domain using our approach and the BM approach. Our approach achieves a better performance with break-even point equal to 0.4, whereas the break-even point of BM approach is 0.28. Recall that the BM approach involves a large amount of human effort in preparing training examples, while our approach is fully automatic. Table 1 depicts the sample output of our product attribute extraction and normalization method in our framework in the camcorder domain. The attribute values are normalized to the appropriate reference attributes. For example, the attribute values corresponding to the reference attribute labeled as "A #1" and "A #2" are related to the power supply and LCD display of a camcorder respectively. Figures 6 and 7 depict the recall-precision plot for the results of product record normalization in the digital camera and MP3 player domains. The BM approach can only correctly normalize a limited number of product records resulting in a low recall. However, our approach is able to automatically discover more normalized product records.

Related Work

Our framework is related to duplicate detection or record linkage in database. Bilenko and Mooney proposed a method which employs learnable edit distance with affine gaps and vector-space model to compute record similarity (Bilenko and Mooney 2003). Sarawagi and Bhamidipaty solved the problem by making use of active learning algorithm to train a combination of similarity functions between attributes of records (Sarawagi and Bhamidipaty 2002). An unsupervised probabilistic method for solving record linkage was proposed by Ravikumar and Cohen (Ravikumar and Cohen 2004). They designed a hierarchical graphical model to model the generation of features of each product pair. Record linkage is achieved by conducting inference for the latent class variables. Culotta et al. proposed a record canonicalization approach which aims at constructing a single canonical form of duplicated records (Culotta et al. 2007). It makes use of the edit distance to compute string similarity and feature based method to select attribute values for the canonical record. All the above methods aim at matching records which have a fixed set of attributes in database. Therefore, they are not applicable to our problem which may involve any number of product attributes and un-

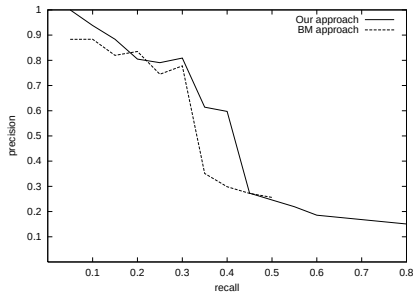

Figure 5: The recall-precision plot for the results of product record normalization in the camcorder domain.

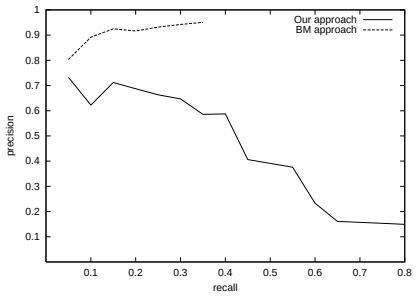

Figure 6: The recall-precision plot for the results of product record normalization in the digital camera domain.

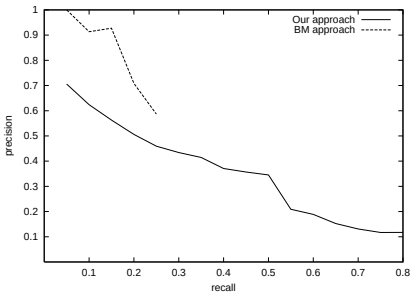

Figure 7: The recall-precision plot for the results of product record normalization in the MP3 player domain.

certain attribute values extracted from Web pages.

The objective of entity resolution shares certain resemblances with our goal. 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 (Singla and Domingos 2006). Bhattacharya and Getoor proposed an unsupervised approach for entity resolution based on Latent Dirichlet Allocation (LDA) (Bhattacharya and Getoor 2006). One limitation of these approaches is that the entities are required to be extracted in advance and cannot be applied to raw data.

Conclusions

We have developed an unsupervised framework for solving the task of product record normalization from different retailer Web sites signifi cantly reducing the human effort. We develop a graphical model to model the generation of text fragments in Web pages. Based on this model, our framework decomposes the product record normalization into two tasks. The fi rst task is the attribute extraction and normalization task. This task aims at extracting the attribute values of product records from different sites, and at the same time normalize them to appropriate reference attributes. The second task is to conduct product record normalization which aims at normalizing product records from different retailer sites to appropriate reference products based on the results of the fi rst task. We have conducted extensive experiments and compared our framework with existing work using over 150 real-world Web sites in three different domains. The experimental results show that our framework can signifi cantly reduce the human effort and achieve a better performance.

References

Bhattacharya, I., and Getoor, L. 2006. A latent dirichlet model for unsupervised entity resolution. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, 47–58.

Bilenko, M., and Mooney, R. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 39–48.

Bilenko, M.; Basu, S.; and Sahami, M. 2005. Adaptive product normalization: Using online learning for record linkage in comparison shopping. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, 58–65.

Blei, D., and Jordan, M. 2006. Variational inference for dirichlet process mixtures. *Bayesian Analysis* 1(1):121–144.

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

Culotta, A.; Wick, M.; Hall, R.; Marzilli, M.; and McCallum, A. 2007. Canonicalization of database records using adaptive similarity measures. In *Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 201–209.

Ishwaran, J., and James, L. 2001. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association* 96(453):161–174.

McCallum, A.; Nigam, K.; and Ungar, L. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 168–178.

Ravikumar, P., and Cohen, W. 2004. A hierarchical graphical model for record linkage. In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI)*, 454–461.

Rurmo, J.; Ageno, A.; and Catala, N. 2006. Adaptive information extraction. *ACM Computing Surveys* 38(2):Article 4.

Sarawagi, S., and Bhamidipaty, A. 2002. Interactive deduplication using active learning. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 269–278.

Sarawagi, S., and Cohen, W. 2004. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems 17, Neural Information Processing Systems*.

Singla, P., and Domingos, P. 2006. Entity resolution with markov logic. In *Proceedings of the Sixth IEEE International Conference on Data Mining*, 572–582.

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

Viola, P., and Narasimhan, M. 2005. Learning to extract information from semi-structured text using a discriminative context free grammar. In *Proceedings of the Twenty-Eighth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 330–337.

Wong, T.-L., and Lam, W. 2007. Adapting web information extraction knowledge via mining site invariant and site depdent features. *ACM Transactions on Internet Technology* 7(1):Article 6.

Wong, T.-L.; Lam, W.; and Wong, T.-S. 2008. 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*, To appear.

Zhao, H.; Meng, W.; and Yu, C. 2007. Mining templates from search result records of search engines. In *Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 884–892.