Unsupervised Extraction of Popular Product Attributes from E-Commerce Web Sites by Considering Customer Reviews

LIDONG BING, Machine Learning Department, Carnegie Mellon University
TAK-LAM WONG, Department of Mathematics and Information Technology, The Hong Kong Institute of Education
WAI LAM, Key Laboratory of High Confidence Software Technologies, Ministry of Education (CUHK Sub-Lab), and Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong

We develop an unsupervised learning framework for extracting popular product attributes from product description pages originated from different E-commerce Web sites. Unlike existing information extraction methods that do not consider the popularity of product attributes, our proposed framework is able to not only detect popular product features from a collection of customer reviews but also map these popular features to the related product attributes. One novelty of our framework is that it can bridge the vocabulary gap between the text in product description pages and the text in customer reviews. Technically, we develop a discriminative graphical model based on hidden Conditional Random Fields. As an unsupervised model, our framework can be easily applied to a variety of new domains and Web sites without the need of labeling training samples. Extensive experiments have been conducted to demonstrate the effectiveness and robustness of our framework.

CCS Concepts: • **Information systems → Extraction, transformation and loading**; **Information extraction;**

Additional Key Words and Phrases: Information extraction, conditional random fields, product attribute, customer reviews

ACM Reference Format:
Lidong Bing, Tak-Lam Wong, and Wai Lam. 2016. Unsupervised extraction of popular product attributes from E-commerce Web sites by considering customer reviews. ACM Trans. Internet Technol. 16, 2, Article 12 (April 2016), 17 pages.
DOI: http://dx.doi.org/10.1145/2857054

12

1. INTRODUCTION

Building intelligent E-commerce systems typically involves a component that can automatically extract product attribute information from a variety of product description pages in different E-commerce Web sites. Web information extraction methods such as

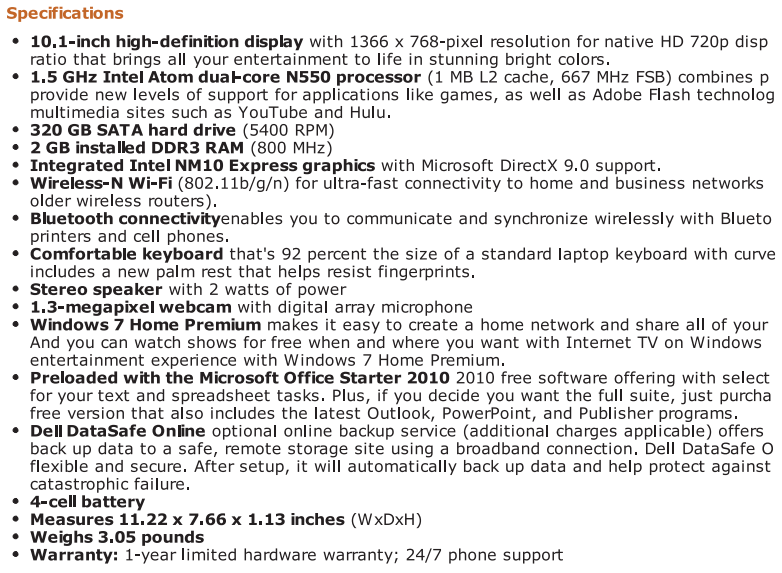
The work described in this article was substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Codes: 413510 and 14203414) and the Direct Grant of the Faculty of Engineering, CUHK (Project Code: 4055034). This work is partially supported by grants from The Hong Kong Institute of Education (Project Codes: RG 18/2014-2015 and RG 30/2015-2016R).
Authors' addresses: L. Bing, Machine Learning Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA; T.-L. Wong, Department of Mathematics and Information Technology, The Hong Kong Institute of Education, 10 Lo Ping Road, Tai Po, N. T., Hong Kong; Contact author's email address: T.-L. Wong, tlwong@ied.edu.hk; W. Lam, Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

DOI: http://dx.doi.org/10.1145/2857054

wrappers are able to automatically extract product attributes from the Web content [Alfonseca et al. 2010; Probst et al. 2007; Zhu et al. 2008]. For example, in the air purifier domain, a product description page may contain a number of product attributes such as filtration, dimension, quietness, and so on. Typically, users need to specify the product attributes of interest in order to select those attributes that are relevant to some features for decision making. It requires a substantial amount of domain knowledge and manual effort. For example, considering again the air purifier domain, a user may consider the feature "filtering capability" as important. Also, this feature is related to the product attribute "filtration." On the other hand, a feature such as "size" may have less influence on decision making. Existing information extraction methods have a common drawback of failing to automatically identify the product attributes in which users are generally interested. Moreover, these kinds of attributes are usually unknown in advance and they vary in different domains.

Opinion mining methods have been investigated and applied to analyze customer review text aiming at extracting product features and detecting opinion orientation [Hu and Liu 2004a, 2004b; Liu et al. 2005; Quan and Ren 2014; Titov and McDonald 2008; Xu et al. 2014]. One may utilize the mining results to compare different products. For example, the feature "filtering capability" may be frequently mentioned in the customer reviews of the air purifier domain. This feature may be associated with the product attribute "filtration." However, the output of opinion mining cannot associate the related attributes due to the gap between the vocabularies used in customer reviews and the terminologies used in product description pages. Users are required to manually map or establish association between the features and product attributes if they want to investigate more on this product or contrast it with others. Such mapping may require some expert knowledge that is far beyond the reach of general users' ability. Consequently, the review mining results cannot help users precisely compare products because the product attributes corresponding to the concerned features are unknown.

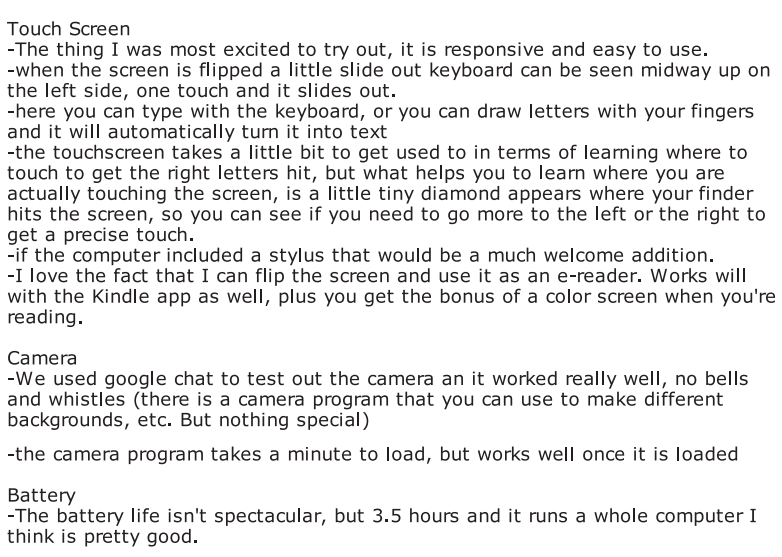
In this article, we develop an unsupervised learning framework for extracting popular product attributes from product description pages originated from different Web sites. Unlike existing systems which do not differentiate the popularity of the attributes, our framework is able to not only detect concerned popular features of a product from a collection of customer reviewsbut also map these popular features to the related product attributes and at the same time extract these attributes from description pages. We explain the rationale of our framework using the following example. Figure 1 shows a Web page about a netbook product. This page contains a list of description such as the text fragments "10.1-inch high-definition display . . . ," "1.5 GHz Intel Atom dual-core N550 processor . . . ," and "2 GB installed DDR3 RAM . . ." showing different attributes of the netbook. However, not all of them are of interest to most customers and therefore cannot influence users' decisions. We wish to extract those attributes which are important for customers to make decisions. To achieve this goal, we make use of a collection of online customer reviews available from Web forums or retailer Web sites as exemplified in Figure 2 to automatically derive the popular features. Note that the concerned product in a Web page is not necessarily contained in the collection of reviews. Each popular feature is represented by a set of terms with weights, capturing the association terms related to that popular feature. For example, the terms like "screen" and "color" are automatically identified to be related to the popular feature "display" of a netbook by analyzing their frequency and co-occurrence in the customer reviews. Our framework can then reinforce that terms like "resolution" and "high-definition" in the text fragments are also related to the popular feature "display." These newly identified terms can be utilized to extract other attributes related to "display" such as "10.1-inch high-definition display." On the other hand, some other



Specifications

- **10.1-inch high-definition display** with 1366 x 768-pixel resolution for native HD 720p disp ratio that brings all your entertainment to life in stunning bright colors.
- **1.5 GHz Intel Atom dual-core N550 processor** (1 MB L2 cache, 667 MHz FSB) combines p provide new levels of support for applications like games, as well as Adobe Flash technolog multimedia sites such as YouTube and Hulu.
- **320 GB SATA hard drive** (5400 RPM)
- **2 GB installed DDR3 RAM** (800 MHz)
- **Integrated Intel NM10 Express graphics** with Microsoft DirectX 9.0 support.
- **Wireless-N Wi-Fi** (802.11b/g/n) for ultra-fast connectivity to home and business networks older wireless routers).
- **Bluetooth connectivity**enables you to communicate and synchronize wirelessly with Blueto printers and cell phones.
- **Comfortable keyboard** that's 92 percent the size of a standard laptop keyboard with curve includes a new palm rest that helps resist fingerprints.
- **Stereo speaker** with 2 watts of power
- **1.3-megapixel webcam** with digital array microphone
- **Windows 7 Home Premium** makes it easy to create a home network and share all of your And you can watch shows for free when and where you want with Internet TV on Windows entertainment experience with Windows 7 Home Premium.
- **Preloaded with the Microsoft Office Starter 2010** 2010 free software offering with select for your text and spreadsheet tasks. Plus, if you decide you want the full suite, just purcha free version that also includes the latest Outlook, PowerPoint, and Publisher programs.
- **Dell DataSafe Online** optional online backup service (additional charges applicable) offers back up data to a safe, remote storage site using a broadband connection. Dell DataSafe O flexible and secure. After setup, it will automatically back up data and help protect against catastrophic failure.
- **4-cell battery**
- **Measures 11.22 x 7.66 x 1.13 inches** (WxDxH)
- **Weighs 3.05 pounds**
- **Warranty:** 1-year limited hardware warranty; 24/7 phone support

Fig. 1. A segment of a description page of a netbook.



Touch Screen
-The thing I was most excited to try out, it is responsive and easy to use.
-when the screen is flipped a little slide out keyboard can be seen midway up on the left side, one touch and it slides out.
-here you can type with the keyboard, or you can draw letters with your fingers and it will automatically turn it into text
-the touchscreen takes a little bit to get used to in terms of learning where to touch to get the right letters hit, but what helps you to learn where you are actually touching the screen, is a little tiny diamond appears where your finder hits the screen, so you can see if you need to go more to the left or the right to get a precise touch.
-if the computer included a stylus that would be a much welcome addition.
-I love the fact that I can flip the screen and use it as an e-reader. Works will with the Kindle app as well, plus you get the bonus of a color screen when you're reading.

Camera
-We used google chat to test out the camera an it worked really well, no bells and whistles (there is a camera program that you can use to make different backgrounds, etc. But nothing special)

-the camera program takes a minute to load, but works well once it is loaded

Battery
-The battery life isn't spectacular, but 3.5 hours and it runs a whole computer I think is pretty good.

Fig. 2. Part of a customer review on a netbook.

attributes, such as “keyboard,” are not mentioned in most reviews. Hence, the text fragment “Comfortable keyboard . . .” will not be extracted. Therefore, our framework can help bridge the vocabulary gap between the features found from the reviews and the attributes in the description pages and extract those attributes related to the popular features. As a result, users can make a wise decision based on the extraction results or search related products based on the popular attributes.

Our proposed framework has several research contributions. The first contribution is that we model the popular attribute extraction as an extraction problem with unknown attributes because the attributes related to popular features are unknown. We

have developed an unsupervised method based on hidden Conditional Random Fields (CRFs) to extract the product attributes from product description pages by considering the terms related to popular features and the layout information of the Web page. The second contribution is the capability of bridging the vocabulary gap between the features found from the reviews and the attributes in the product description Web pages. Popular attributes can also be extracted in our framework. Third, we have conducted extensive experiments on a large number of product description pages that are collected from 13 different domains. We have also conducted comparisons with some existing models that can solve this unsupervised popular attribute extraction problem in a reasonable manner. The experimental results can demonstrate the effectiveness and robustness of our framework.

This article substantially extends our previous work in Bing et al. [2012]. First, we elaborate on more technical details of the proposed model, such as dynamic-programming-based inference and lower-bound-based unsupervised learning. Second, more experiments are conducted to validate the effectiveness of our method. Specifically, the number of the experimental domains is almost doubled and another comparison method is implemented. In addition, some case studies are also given. Third, the difference between our work and previous works is discussed whenever appropriate in different sections, such as the discussion on the application aspects in Sections 1 and 7, the discussion on the modeling aspect in Section 4, and so on.

The remainder of the article is organized as follows. We first give the problem definition in Section 2. After the overview of the proposed framework is described in Section 3, the details of its two major components are discussed in Section 4 and Section 5, respectively. The experimental setup and results are given in Section 6. After some related works are reviewed in Section 7, we conclude the article and provide some possible directions for the future work in Section 8.

2. PROBLEM DEFINITION

We use two key notions, namely feature and attribute. Both feature and attribute refer to an aspect characterizing a product of a particular domain. We use "attribute" to refer to such aspect in the product description Web pages and use "popular feature" to refer to the discovered hidden aspects/concepts from the reviews. Our model automatically builds correlations between the attributes and the popular features so the popular attributes are identified. Therefore, we do not make the assumption that each discovered feature from the reviews corresponds to a popular attribute.

In a particular domain, let $\boldsymbol{A} = \{A_1, A_2, \ldots\}$ be the set of underlying attributes characterizing the products in this domain. For example, the set of product attributes of the netbook domain includes "screen," "multi-media," and so on. Given a Web page W about a certain product in the given domain, W can be treated as a sequence of tokens $(tok_1, \ldots, tok_{N(W)})$, where $N(W)$ refers to the number of tokens. We also define $tok_{l,k}$ as a text fragment composed of consecutive tokens between tok_l and tok_k in W, where $1 \leq l \leq k \leq N(W)$. Let $L(tok_{l,k})$ and $C(tok_{l,k})$ be the layout features and the content features of the text fragment $tok_{l,k}$, respectively. We denote $V(tok_{l,k}) = A_j$ if $tok_{l,k}$ is related to the attribute A_j.

We denote $\boldsymbol{A_{POP}} \subseteq \boldsymbol{A}$ as the set of popular product attributes that are of our interest in this task. Note that $\boldsymbol{A_{POP}}$ is related to the popular features $\boldsymbol{C(R)}$, discovered from a collection of customer reviews $\boldsymbol{R}$ about some products in the same domain. Our popular attribute extraction problem can be defined as follows: Given a Web page W of a certain product and a set of customer reviews $\boldsymbol{R}$ in the same domain, we aim at automatically identifying all the possible text fragments $tok_{l,k}$ in W such that $V(tok_{l,k}) = A_j$ and $A_j \in \boldsymbol{A_{POP}}$, by considering $L(tok_{l,k})$, $C(tok_{l,k})$, and the popular features $\boldsymbol{C(R)}$. Note that $\boldsymbol{A_{POP}}$ is automatically derived from $\boldsymbol{C(R)}$ beforehand and does not need to be

pre-specified in advance. In addition, the concerned product in the page W does not necessarily appear in $\boldsymbol{R}$.

3. OVERVIEW OF OUR FRAMEWORK

Our proposed framework is composed of two major components. The first component is the popular attribute extraction component, which aims at extracting text fragments corresponding to the popular attributes from the product description Web pages. Web pages are regarded as a kind of semi-structured text documents containing a mix of structured content such as HTML tags and free texts which may be ungrammatical or just composed of short phrases. Given a Web page W about a certain product in the given domain as a sequence of tokens $(tok_1, \ldots, tok_{N(W)})$, our goal is to identify all text fragments $tok_{l,k}$ such that $V(tok_{l,k}) = A_j$ and $A_j \in \boldsymbol{A_{POP}}$ where $\boldsymbol{A_{POP}} \subseteq \boldsymbol{A}$. This task can be formulated as a sequence labeling problem. Precisely, we label each token in $(tok_1, \ldots, tok_{N(W)})$ with two sets of labels. The first set contains the labels "B," "I," and "O" denoting the beginning of an attribute, inside an attribute, and outside an attribute, respectively. The second set of labels are $A_j \in \boldsymbol{A_{POP}}$, that is, the type of popular attributes. CRFs have been adopted as the state-of-the-art model to deal with sequence labeling problems. However, existing standard CRF models are inadequate to handle this task for several reasons. The first reason is that each token will be labeled by two kinds of labels simultaneously, whereas standard CRFs only consider one kind of label. The second reason is that the popular attributes are related to the hidden concepts derived from the customer reviews by the second component and are unknown in advance. This leads to the fact that supervised training adopted in standard CRFs cannot be employed. To tackle this problem, we have developed a graphical model based on hidden CRFs. The proposed graphical model can exploit the derived hidden concepts, as well as the clues from layout features and text content features. An unsupervised learning algorithm is also developed to extract the popular attributes.

The second component aims at automatically deriving $\boldsymbol{A_{POP}}$ from a collection of customer reviews $\boldsymbol{R}$. This component first generates a set of derived documents from $\boldsymbol{R}$. Latent Dirichlet Allocation (LDA) [Blei et al. 2003] is then employed to discover latent concepts, which essentially refer to the popular features of the products $\boldsymbol{C(R)}$, from the derived documents. Each $c \in \boldsymbol{C(R)}$ is essentially represented by a multinomial distribution of terms. For example, one popular feature is more likely to generate the terms "display," "resolution," "screen," and so on, while another popular feature is more likely to generate the terms "camera," "speaker," and so on. By making use of such information on terms, our graphical model can extract the text fragments related to the popular attributes.

4. POPULAR ATTRIBUTE EXTRACTION METHOD

4.1. Hidden CRF Model

Figure 3 shows the graphical model capturing the inter-dependency among the essential elements in the extraction problem. Each node and edge of the graphical model represents a random variable and the dependence between two connected nodes. We first conduct some simple preprocessing by analyzing the DOM structure to decompose a Web page into a sequence of tokens $(tok_1, \ldots, tok_{N(W)})$. Specifically, the text context of a page is extracted by traversing the DOM tree with pre-order traversal. In addition, we extract some layout features from the DOM tree, such as the font information of each token and whether the current sentence is an item in a list. Let the random variable $\boldsymbol{X}$ refer to the observation from the sequence and incorporate the orthographical information of the tokens or the layout format of the Web page. Another set of random variables denoted as $\boldsymbol{Y} = (y_1, \ldots, y_{N(W)})$ ranging over a finite set of label alphabet $\mathcal{Y}$

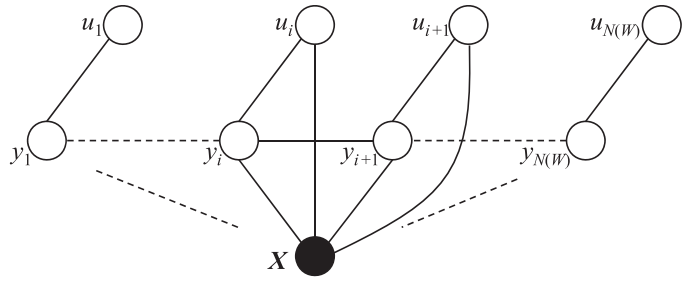


Fig. 3. The graphical model for popular product attribute extraction. (Note that all u and y are connected to X in the model. Some obvious links are not shown in the diagram for clarity.)

refer to the class labels of each token. Recall that each $tok_{l,k}$ corresponds to a contiguous text fragment between tok_l and tok_k. Hence, each y_i can be equal to "B," "I," or "O" denoting the beginning of an attribute, inside an attribute, and out of an attribute, respectively. In order to incorporate the information of the derived hidden concepts, which represent the popular product attributes discovered from the customer reviews, we design another set of random variables $\boldsymbol{U} = (u_1, \ldots, u_{N(W)})$ ranging over $\boldsymbol{A_{POP}} \cup \{\bar{A}\}$, where $\bar{A}$ is a special symbol denoting "not-a-popular-attribute." Essentially, each u_i represents the popular attribute that tok_i belongs to. We use $\boldsymbol{V}$, $\boldsymbol{E^Y}$, and $\boldsymbol{E^U}$ to denote the set of all vertices, the set of all edges connecting two adjacent ys, and the set of all edges connecting a particular y and a particular u, respectively. Li et al. [2009] have also proposed a CRF-based model with derived labels to tackle the query tagging problem. However, our graphical model described above has a different graphical structure since the problems to be solved differ. Moreover, in our hidden CRF model, the labels of the layers $\boldsymbol{Y}$ and $\boldsymbol{U}$ are of different types.

Our model is in the form of a linear chain. Hence, the joint distribution $P_\theta(\boldsymbol{Y} = y, \boldsymbol{U} = u|\boldsymbol{X} = x)$ over the class label sequence y and the popular attribute labels u given the observation x and the set of parameters θ can be expressed as follows by the Hammersley-Clifford theorem:

$$P_\theta(\boldsymbol{Y} = y, \boldsymbol{U} = u|\boldsymbol{X} = x) = \frac{1}{Z(x)} \exp\left\{ \sum_{e \in E^Y, k} \lambda_k f_k(e, y|_e, x) + \sum_{v \in V, k} \mu_k g_k(v, y|_v, x) + \sum_{e \in E^U, k} \gamma_k h_k(e, y|_e, u|_e, x) \right\}, \quad (1)$$

where $f_k(e, y|_e, x)$ refers to the feature function related to x, the nodes ys connected by the edge $e \in \boldsymbol{E^Y}$; $g_k(v, y|_v, x)$ refers to the feature function related to x, the node v represented by the vertex $v \in \boldsymbol{V}$; $h_k(e, y|_e, u|_e, x)$ refers to the feature function related to x, the nodes u and y connected by the edge $e \in \boldsymbol{E^U}$; and λ_k, μ_k, γ_k are the parameters associated with $f_k(e, y|_e, x)$, $g_k(v, y|_v, x)$, and $h_k(e, y|_e, u|_e, x)$ respectively. $Z(x)$, which is a function of x, is the normalization factor. As a result, the goal of our popular attribute text fragment extraction is to find the labeling of y and u given the sequence x and the model parameter θ which includes all the λ_k, μ_k, and γ_k, such that $P_\theta(\boldsymbol{Y} = y, \boldsymbol{U} = u|\boldsymbol{X} = x)$ is maximized.

4.2. Inference

For simplicity, we use $P_\theta(y, u|x)$ to replace $P_\theta(\boldsymbol{Y} = y, \boldsymbol{U} = u|\boldsymbol{X} = x)$ when the context is clear. Moreover, we will follow the notation in Lafferty et al. [2001] to describe our method. We add the special labels "*start*" and "*end*" for y_0 and $y_{N(W)+1}$ for easy

illustration of our method. Recall that our goal is to compute $\arg\max_{y,u} P_\theta(y, u|x)$, which can be expressed as Equation (1). For each token tok_i in a sequence, we define the following $|\boldsymbol{Y}| \times |\boldsymbol{U}|$ matrix:

$$\Lambda_i^U(y, u|x) = \sum_k \gamma_k h_k(e_i, y|_{e_i}, u|_{e_i}, x). \tag{2}$$

We then can define the following $|\boldsymbol{Y}| \times |\boldsymbol{Y}|$ matrices:

$$\Lambda_i^Y(y', y|x) = \sum_k \lambda_k f_k(e_i, y|_{e_i}, x) + \sum_k \mu_k g_k(v_i, y|_{v_i}, x) + \sum_{u'} \Lambda_i^U(y, u'|x) \tag{3}$$

and

$$M_i(y', y|x) = \exp(\Lambda_i^Y(y', y|x)). \tag{4}$$

Given the above matrices, $P_\theta(y, u|x)$ for a particular y and u given x can then be computed as follows:

$$P_\theta(y, u|x) = \frac{\prod_{i=1}^{N(W)+1} M_i(y', y|x)\left[\frac{\exp(\Lambda_i^U(y,u|x))}{\exp\left(\sum_{u'} \Lambda_i^U(y,u'|x)\right)}\right]}{Z(x)}, \tag{5}$$

where $Z(x) = (\prod_{i=1}^{N(W)+1} M_i(y', y|x))_{start,end}$ is the normalization factor.

During inference, we are given the sequence of tokens and its observation. We define the forward vectors α_i and the backward vectors β_i for each tok_i as follows:

$$\begin{aligned} \alpha_0(y|x) &= \begin{cases} 1 & \text{if y} = start \\ 0 & \text{otherwise} \end{cases} \\ \alpha_i(x) &= \alpha_{i-1}(x) M_i(x), \end{aligned} \tag{6}$$

$$\begin{aligned} \beta_{N(W)+1}(y|x) &= \begin{cases} 1 & \text{if y} = end \\ 0 & \text{otherwise} \end{cases} \\ \beta_i(x)^T &= M_{i+1}(x)\beta_{i+1}(x). \end{aligned} \tag{7}$$

Next, we can compute the optimal labeling of y and u using dynamic programming. In particular, the optimal labeling for y_i and u_i are expressed as follows:

$$\begin{aligned} (u_i^*, y_i^*) &= \arg\max_{y',u'} P(y_i = y', u_i = u') \\ &= \arg\max_{y',u'} \frac{\alpha_i(y'|x)\beta_i(y'|x)}{Z(x)}\left[\frac{\exp(\Lambda_i^U(y', u'|x))}{\exp\left(\sum_{u''} \Lambda_i^U(y', u''|x)\right)}\right]. \end{aligned} \tag{8}$$

Our proposed hidden CRF model significantly differs from the ones reported in Quattoni et al. [2007] and Sung and Jurafsky [2009] applied in speech recognition and object recognition, respectively. From the modeling perspective, both Quattoni et al. [2007] and Sung and Jurafsky [2009] propose a CRF model with one hidden layer between the observation layer and the label layer. The hidden layer is mainly used to enhance the model structure and does not carry explicit semantic meaning. On the contrary, the hidden nodes $\boldsymbol{Y}$ and $\boldsymbol{U}$ in our model embody extraction information and popular product attribute information, respectively. From a technical perspective, Quattoni et al. [2007] and Sung and Jurafsky [2009] consider the conditional probability $P(\boldsymbol{Y}|\boldsymbol{X})$, where $\boldsymbol{U}$ is marginalized. However, our model considers the joint conditional probability $P(\boldsymbol{Y}, \boldsymbol{U}|\boldsymbol{X})$. Moreover, $\boldsymbol{Y}$ is known in the training of Quattoni et al. [2007] and Sung and Jurafsky [2009], whereas both $\boldsymbol{Y}$ and $\boldsymbol{U}$ are unknown in our

model. During training, Quattoni et al. [2007] and Sung and Jurafsky [2009] aim at maximizing the likelihood function for generating the labels of the hidden nodes $\boldsymbol{Y}$ of the training examples. However, we develop a training algorithm that does not require the labels for both hidden nodes $\boldsymbol{Y}$ and $\boldsymbol{U}$.

4.3. Unsupervised Learning

We have developed an unsupervised method for learning our hidden CRF model, given a set of M unlabeled data $\mathcal{D}$, in which the observation $\boldsymbol{X}$ of each sequence is known, but the labels y and u for each token are unknown. In principle, discriminative learning is impossible for unlabeled data. To address this problem, we make use of the customer reviews to discover a set of hidden concepts and predict a derived label for u of each token. Note that the derived labels are just used in the learning and they are not used in the final prediction for the unlabeled data. As a result, we can exploit the derived label u and the observation $\boldsymbol{X}$ in learning the model. The approach of discovering hidden concepts will be described in the next section.

Since the class label y of each token is unknown, we aim at maximizing the following log-likelihood function in our learning method:

$$\begin{aligned}\mathcal{L}_\theta &= \sum_{m=1}^{M} \log P(u^{(m)}|x^{(m)};\theta) \\ &= \sum_{m=1}^{M} \log \sum_{y'} P(y', u^{(m)}|x^{(m)};\theta). \end{aligned} \tag{9}$$

Maximizing this log-likelihood function is intractable because of the summation with the logarithm function. To address this, we obtain the following inequality according to Jensen's inequality and the concave property of the logarithm function:

$$\begin{aligned}\mathcal{L}'_\theta &= \sum_{m=1}^{M} \sum_{y'} \log P(y', u^{(m)}|x^{(m)};\theta) \\ &= \sum_{m=1}^{M} \sum_{y'} \left\{ \sum_{e\in E^Y,k} \lambda_k f_k(e, y'|_e, x^{(m)}) + \sum_{v\in V,k} \mu_k g_k(v, y'|_v, x^{(m)}) \right. \\ &\quad \left. + \sum_{e\in E^U,k} \gamma_k h_k(e, y'|_e, u^{(m)}|_e, x^{(m)}) \right\}. \end{aligned} \tag{10}$$

Note that Equation (10) is the lower bound of Equation (9). Consequently, instead of directly maximizing Equation (9), our learning method aims at maximizing its lower bound depicted in Equation (10). By taking the first derivative of Equation (10) with respective to each parameter and setting to zero, we can obtain the optimal condition. In particular, the first partial derivatives with respective to λ_k, μ_k, and γ_k are shown as follows:

$$\frac{\partial \mathcal{L}'_\theta}{\partial \lambda_k} = \sum_{m=1}^{M} \sum_{y'} f_k(e, y'|_e, x^{(m)}) - \sum_{m=1}^{M} \sum_{y'} Exp_{p_\theta(y', u|x^{(m)})}[f_k(e, y'|_e, x^{(m)})], \tag{11}$$

$$\frac{\partial \mathcal{L}'_\theta}{\partial \mu_k} = \sum_{m=1}^{M} \sum_{y'} g_k(v, y'|_v, x^{(m)}) - \sum_{m=1}^{M} \sum_{y'} Exp_{p_\theta(y', u|x^{(m)})}[g_k(v, y'|_v, x^{(m)})], \tag{12}$$

$$\frac{\partial \mathcal{L}'_\theta}{\partial \gamma_k} = \sum_{m=1}^{M} \sum_{y'} h_k(e, y'|_e, u^{(m)}|_e, x^{(m)}) - \sum_{m=1}^{M} \sum_{y'} Exp_{p_\theta(y', u|x^{(m)})}[h_k(e, y'|_e, u^{(m)}|_e, x^{(m)})]. \quad (13)$$

As a result, we can employ efficient optimization algorithms such as the limited memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm to compute the optimal parameter sets.

One issue of this formulation method is that the function shown in Equation (10) is not concave. The optimal condition obtained may be a local optimum. The obtained solution is affected by the starting point of the BFGS algorithm. To improve the quality of the result, we can initialize the algorithm with carefully constructed starting points. We observe that most popular product attribute values are of noun phrases in the product description Web pages, such as "good odor reduction" and "quiet and quick clean fan mode." As a result, initializing the parameters of the features, which are associated with noun phrases, with higher values is useful for achieving a better performance. For example, the feature weight μ_k for $g_k(v, y|_v, x)$ will be set to a higher value if x refers to the observation that the part of speech of the underlying token is a noun.

5. POPULAR FEATURE IDENTIFICATION FROM CUSTOMER REVIEWS

In this section we present our method for discovering hidden concepts from a collection of customer reviews. We observe that most customer reviews are organized in paragraphs as exemplified by the reviews in Figure 2. The customers usually described a few related aspects in one paragraph and mentioned less related aspects in different paragraphs when they post a multi-paragraph review. To facilitate the discovery of high-quality hidden concepts, we treat each paragraph as a processing document unit in our hidden concept discovery method. After the preprocessing operations such as stemming and stop word removal are conducted, a derived document is obtained from each review paragraph. Finally, the collection of derived documents for the review collection $\boldsymbol{R}$ can be obtained. We employ LDA to discover the hidden concepts, which essentially refer to the popular features, for a domain.

6. EXPERIMENTS AND DISCUSSIONS

6.1. Data Preparation

We have conducted extensive experiments to evaluate the performance of our framework. We collected a large number of product description pages from over 20 different online retailer Web sites covering 13 different domains. Table I shows the domain label, the domain name, the number of different products of each domain, and the number of product description pages of each domain. In addition, we have collected more than 500 customer reviews in each domain similar to the one shown in Figure 2 from retailer Web sites. These reviews were fed into the hidden concept discovery algorithm described in Section 5 and the number of latent topics was set to 30 for each domain. We filtered out those output latent topics dominated by the terms which are obviously not related to product description terms, such as "husband," "son," "Amazon," and so on. Typically, it is easy to identify such kind of latent topics to filter out. Such filtering work is only a one-time small amount of manual effort for a given domain. The remaining latent topics are directly utilized as the derived concepts in the hidden CRF model.

Three annotators were hired to identify the popular product attribute text fragments from the product description pages for evaluation purpose. Each annotator first read

Table I. The Details of the Data Collected for the Experiments

Domain Label	Domain Name	# of different products	# of pages
D1	baby car seat	15	61
D2	bread maker	13	47
D3	carpet cleaner	16	48
D4	digital camera	19	86
D5	disc player	15	47
D6	GPS device	17	57
D7	LCD TV	16	47
D8	netbook	15	44
D9	phone	14	41
D10	printer	16	47
D11	purifier	15	44
D12	scanner	16	48
D13	watch	15	46

through the reviews by herself. She stopped reading when shefelt confident that she already had a good understanding of the popular features of the domain, as she had read at least 200 reviews. We generated three random lists of those reviews and each of them was given to one annotator to avoid the bias of the original review order on the annotators' judgment of popular features. Then they discussed and determined the final set of popular features, as well as some sample popular product attributes for the domain. After that, the annotators used the information to guide their annotation work of the corresponding domain, and the text fragments corresponding to popular attributes were manually identified by them individually from product description pages. Finally, the text fragments that were identified by at least two annotators were kept as true popular attributes. We assessed the agreement level among the annotators with Fleiss's Kappa [Fleiss 1971]. The whole set of annotation subjects includes two parts, namely, the identified distinct text fragments by annotators and the noun phrases in the description that do not overlap with the annotated fragments. The calculated Kappa value is 0.638, which suggests a substantial agreement level.

6.2. Experimental Setting

Since there is no existing unsupervised method that extracts popular product attributes from pages by considering customers' interest revealed by reviews, we implemented two comparison methods based on integration of some existing methods.

The first comparison method is called "VIPS-Bayes," which consists of two steps. The first step is to extract the text fragments of related product attributes from Web pages. The second step is to determine the popular attributes. For the first step, we first conduct unsupervised Web data extraction based on VIPS, which is an existing automatic Web page segmentation method for segmenting a page into logical blocks [Cai et al. 2004]. Since we have observed that almost all of the popular product attribute values (text fragments) are noun phrases, we apply the openNLP[1] package to conduct noun phrase extraction from the text in the product description blocks. The identified noun phrases, in which each of the terms is stemmed, become the popular attribute value candidates. In the second step, we determine the popular attribute values as follows. We discover the derived hidden concepts for a domain using LDA from the customer reviews with the same method as discussed in Section 5. Note that each hidden concept is represented by a set of terms with probabilities. The probability refers to how likely it is that a term is generated from a particular derived hidden concept. Next, each popular attribute value candidate is scored using Bayes's theorem

[1]http://opennlp.sourceforge.net/.

and independence assumption. The score is defined as the conditional probability that the candidate comes from a particular derived hidden concept given the set of terms contained in the candidate. Specifically, it is calculated as the normalized product of the probabilities of the terms belonging to a particular concept. Those candidates with scores greater than a certain threshold will be considered as popular attribute values. For each domain, we invoke a parameter tuning process. The aim is to determine the optimal threshold value so the best performance of VIPS-Bayes is achieved with the F1-measure as the metric.

The second comparison method, namely, "VIPS-Opinion Observer," is designed based on an existing opinion mining method. It first extracts the noun phrases similar to "VIPS-Bayes" and obtains the popular attribute candidates. Next, we utilize the collection of reviews to score these candidates by implementing an existing opinion mining method called Opinion Observer [Hu and Liu 2004b; Liu et al. 2005]. Precisely, we implement their important feature extraction method, which can deal with free format reviews, that is, format 3 mentioned in Hu and Liu [2004b]. For each noun phase candidate found in the first step, if its term set, obtained after stemming and stop word removal, is equal to the term set of an important feature extracted from the reviews, then it becomes a popular attribute value. The parameter setting described in Hu and Liu [2004b] is adopted in our implementation. Specifically, the minimum support and the $p - support$ are set to 1% and 3, respectively.

6.3. Quantitative Results

Evaluation is conducted by comparing the manual annotated answers with the output of the systems. We adopt the standard precision and recall metrics. Precision is defined as:

$$P = \frac{TP}{(TP + FP)}, \tag{14}$$

where TP is the number of extracted popular attribute text fragments that are correct for a particular page, and FP is the number of extracted text fragments that are wrong. Recall is defined as:

$$R = \frac{TP}{(TP + FN)}, \tag{15}$$

where FN is the number of true popular attributes annotated by the annotators but not extracted by the system. We also calculate the F1-measure, which is defined as the harmonic mean of precision and recall with equal weight:

$$F = \frac{2PR}{P + R}. \tag{16}$$

After calculating the precision, recall, and F1-measure values for each page in a particular domain, the corresponding macro-averaged value across all pages in this domain is calculated, and this value is reported as the performance for this domain.

Table II depicts the extraction performance of each domain and the average extraction performance among all domains of our approach, the "VIPS-Bayes" and "VIPS-Opinion Observer." It can be observed that our approach achieves the best performance. The average F1-measure of our approach is 0.702, while the average F1-measure values of "VIPS-Bayes" and "VIPS-Opinion Observer" are 0.569 and 0.568 respectively. In addition, paired t-tests (with $P < 0.001$) comparing our approach with the other two approaches show that the performance of our approach is significantly better. It illustrates that our approach can leverage the clues to make coherent decisions in both the product attribute extraction task and the popular product attribute classification

Table II. The Popular Attribute Extraction Performance of Our Approach and Comparison Methods. P, R, and F1 Refer to the Precision, Recall, and F1-Measure, Respectively

Domain	Our Approach			VIPS-Bayes			VIPS-Opinion Observer		
	P	R	F1	P	R	F1	P	R	F1
D1	0.692	0.774	0.722	0.530	0.778	0.615	0.444	0.725	0.542
D2	0.570	0.787	0.706	0.795	0.485	0.582	0.435	0.817	0.520
D3	0.641	0.880	0.729	0.462	0.915	0.597	0.485	0.888	0.610
D4	0.789	0.706	0.735	0.644	0.740	0.673	0.658	0.727	0.680
D5	0.567	0.859	0.678	0.388	0.819	0.513	0.386	0.845	0.519
D6	0.624	0.793	0.681	0.377	0.820	0.499	0.319	0.767	0.442
D7	0.635	0.784	0.716	0.443	0.782	0.502	0.442	0.817	0.542
D8	0.769	0.802	0.762	0.709	0.796	0.742	0.575	0.850	0.676
D9	0.731	0.829	0.769	0.706	0.773	0.733	0.639	0.833	0.716
D10	0.561	0.758	0.624	0.384	0.839	0.505	0.346	0.760	0.462
D11	0.620	0.684	0.662	0.581	0.332	0.367	0.470	0.709	0.554
D12	0.587	0.802	0.671	0.470	0.919	0.610	0.756	0.502	0.583
D13	0.584	0.729	0.670	0.562	0.453	0.467	0.463	0.788	0.539
Avg.	0.644	0.784	0.702	0.543	0.727	0.569	0.494	0.771	0.568

task, leading to a better overall performance. One major reason for the difference is due to the vocabulary gap between the description page and the customer reviews as described in Section 1. In our approach, hidden concepts, represented by a distribution of terms, can effectively capture the terms related to a popular attribute. As the hidden concept information, together with the content information and the layout information of each token, are utilized, our hidden CRF model can accurately extract the popular attribute text fragments from description pages. Another observation is that the precision of our approach is significantly higher because our hidden CRF model utilizes the customer reviews in a more delicate manner, that is, the derived concepts which are concerned by the customers. Although the VIPS-Bayes method also employs the derived concepts to identify the popular attributes, it is unable to apply the rich features such as layout features. In addition, our hidden CRF model performs sequential labeling on the token sequence, which is more robust than the ad hoc manner of VIPS-Bayes. VIPS-Opinion Observer is not able to identify the popular attributes containing different terms as used in the reviews, since it matches the candidate popular attributes with the important attributes extracted from the reviews.

As mentioned in Section 4.3, for alleviating the effect of local optimum problem, we attempt to pick better starting point for the BFGS algorithm. Specifically, the weight parameters will be set to 1 if the corresponding x of those features refers to a token of noun. In addition, we invoke the learning by using different random initial values in the range of $[-0.5, 0.5]$ for other model parameters. The result of each domain reported in Table II is from the run that achieves the largest value for Equation (10) among 10 runs. To verify the efficacy of our parameter initialization strategy, we also evaluate our framework under a totally random strategy for initializing the parameters. Ten runs are invoked for each domain, and the result of the run achieving the largest value for Equation (10) is picked as the result of this domain. Under the random strategy, the averaged F1 value of these 13 domains is 0.665. Therefore, the proposed initialization strategy achieves about 5.5% improvement.

6.4. Qualitative Results

Table III shows five samples of popular product attribute text fragments extracted by our framework for each of three different products from the purifier domain. For example, in the purifier domain, the text fragments of the attribute "quietness" are "the quiet fan" and "a quiet operation," and the text fragments of the attribute "filtrates"

Table III. The Five Samples of Extracted Popular Attribute Text Fragments Extracted by Our Framework for Three Products in the Purifier Domain

Product ID	Extracted Popular Attributes
Product 1	a two year warranty electronic filter replacement indicator long filter life the quiet fan the TrueAir plug-mount odor eliminator
Product 2	125 square feet room a control ionize a quiet operation dust pollen and pet odor micro size air particle
Product 3	easy replacement good odor reduction medium room quick clean mode quiet and quick clean fan mode

Table IV. The Top Five Weighted Terms in Five Hidden Concepts Discovered from the Customer Reviews in the LCD-TV Domain Using Our Framework

Concept 1	Concept 2	Concept 3	Concept 4	Concept 5
picture	wall	sound	color	remote
clear	back	speaker	black	button
quality	stand	audio	contrast	sensitive
hdtv	mount	system	absolute	control
image	base	volume	lightness	menu

are “dust pollen and pet odor” and “micro size air particle.” It can be observed that the text fragments that appeared in the product description pages belong to some popular product attributesthat are frequently mentioned in the customer reviews. As a result, these product attributes can effectively contrast the product from others. Such information is very helpful for customers to make purchase decisions, as well as for retailers and manufacturers to promote their products.

Table IV shows the top five weighted terms in five different concepts discovered in the LCD-TV domain. It can be observed that the semantic meaning of the concepts can be easily interpreted. For example, the five concepts in the LCD-TV domain is related to the popular attributes picture quality, wall mount, sound quality, color, and ease of use, respectively. These attributes are considered to be important to customers because their associated terms are frequently mentioned in the customer reviews. Therefore, differing from existing extraction methods, our framework considers the relevance between the interest of consumers and the extracted popular product attributes.

Table V shows samples of popular product attribute text fragments extracted by our framework for three different products in the netbook domain. Each extracted text fragment is related to a popular feature depicted in the first row of the table. The second row of the table shows the top-five weighted terms in the popular features identified from the customer reviews. These attribute text fragments actually appear in the product description pages and are considered important by the customers, and they are semantically related to the corresponding features discovered from the reviews. In addition, we can see that some attributes, such as “normal tft lcd,” show significant vocabulary gaps with the features and our framework can successfully extract them.

Table V. Samples of Extracted Popular Attribute Text Fragments Associated with Three Different Popular Features Extracted by Our Framework in the Netbook Domain

	Concept 1	Concept 2	Concept 3
	wireless, network, wi-fi, internet, connection	screen, resolution, display, brightness, monitor	battery, life, 6-cell, charge, capacity
Product 1	"draft-n wi-fi network," "the integrated bluetooth connectivity"	"10.1 inches ultrawide display," a superbright 10.1-inch lcd	"extended battery life," "the traditional lithium battery"
Product 2	"fast ethernet connection," "the bluetooth 2.0+edr technology"	"normal tft lcd," "10.1 inches widescreen"	"65 hours battery life," "a 6 cells battery"
Product 3	"ultimate wireless accessibility," "wired ethernet lan"	"widescreen trubrite display," "a 1024 × 600-pixel resolution"	"long battery life," "6-cell lithium-ion battery"

The second row shows the top-five weighted terms in the popular features discovered from the customer reviews. Each cell of the table contains the popular attribute text fragments extracted from product description pages.

6.5. Application Discussions

The extracted popular attributes can be utilized in both practical application scenarios and other research. For practical application, one straightforward way is to highlight the popular attributes in the product description pages, so customers can easily find the attributes in which they may be interested. Another way is that the retailers can design popular-attribute-oriented promotion plans, because such plans are more likely able to attract the attention of potential customers and result in purchases. The third direction is that researchers may investigate reorganizing the reviews according to the extracted popular attributes. For example, the reviews related to a popular attribute are presented in a popup frame when the customers click or move the mouse over the attribute. Thus, the customers can easily obtain the feedback of other customers regarding this attribute. The fourth direction is that our framework can be extended to incorporate other types of reader feedback information, such as microblogs, that are found useful for product recommendation [Zhao et al. 2014].

7. RELATED WORK

Automated information extraction from Web pages has drawn much attention. Some approaches rely on wrappers which can be automatically constructed via wrapper induction. For example, Zhu et al. [2008] developed a model known as Dynamic Hierarchical Markov Random Fields (DHMRF), which is derived from Hierarchical CRFs (HCRF). Their DHMRF model is able to integrate data record detection and attribute labeling. Zheng et al. [2009] proposed a method for extracting records and identifying the internal semantics at the same time. Yang et al. [2010] developed a model combing HCRF and semi-CRF that can leverage the Web page structure and handle free texts for information extraction. Luo et al. [2009] studied the mutual dependencies between Web page classification and data extraction and proposed a CRF-based method to tackle the problem. Wong and Lam [2007] aimed at reducing the human work of preparing training examples by automatically adapting extraction knowledge learned from a source Web site to new unseen sites and discover new attributes. Some common disadvantages of the above methods are that human effort is needed to prepare training examples and the attributes to be extracted are pre-defined. These shortcomings will lead to poor performance for new domains or Web sites. In this article, we propose an unsupervised framework which requires no training examples and does not need to pre-define the product attributes.

Some Web data extraction approaches do not need labeled data. Mining Data Records (MDR) method [Liu et al. 2003] and Record Segmentation Tree (RST) method [Bing

et al. 2011] were developed to automatically detect data records in a Web page, and these records are further processed to extract record attributes [Zhai and Liu 2006; Bing et al. 2013]. Song et al. [2010] developed a technique called MiDAT on the basis of DEPTA, which can extract the data records that contain user-generated content by applying certain domain constraints. The above two algorithms require the existence of a certain Web page format, such as the presence of a list of records or the existence of some domain constraints. Wong et al. [2008, 2011] proposed learning methods for attribute extraction and normalization. We aim at extracting product attributes that are of interests to consumers.

Some existing methods have been developed for information extraction of product attributes based on text mining. Ghani et al. [2006] proposed employing a classification method for extracting attributes from product description texts. Probst et al. [2007] proposed a semi-supervised algorithm to extract attribute value pairs from text descriptions. Their approach aims at handling free text descriptions by making use of natural language processing techniques. Hence, it cannot be applied to Web documents that are composed of a mix of HTML tags and free texts. Although the above methods also extract attributes from the product description texts, they do not exploit the text content of the reviews from customers.

The goal of extracting popular product attributes from product description Web pages differs from opinion mining or sentiment detection research as exemplified in Ding et al. [2009], Liu et al. [2005], Tang et al. [2009], and Turney [2002]. Some methods need domain knowledge [Bloom et al. 2007] or additional semantic resources, such as employing a Web search engine [Popescu and Etzioni 2005]. Some need pre-defined patterns to extract the attributes formatted with these patterns [Kobayashi et al. 2004]. Hu et al. developed a method for extracting the attributes in reviews by mining frequent itemsets [Hu and Liu 2004a, 2004b; Liu et al. 2005] without additional manual effort. In Zhang et al. [2010], an extraction method based on double propagation was proposed. This method extracts all the attributes in the reviews (frequent and infrequent ones) and needs a manually generated stop word list for the "no" pattern. Wang et al. [2010] proposed a technique that can discover latent aspects from review texts. Guo et al. [2009] developed a method for product attribute extraction and categorization from semi-structured customer reviews based on latent semantic association. These methods typically discover and extract all product attributes as well as opinions directly appeared in customer reviews. These methods focus on detecting sentiment terms. In contrast, our goal is to discover popular product attributes from description Web pages. Due to different goals and nature of the texts, techniques used in opinion mining or sentiment detection are unable to fulfill the extraction task investigated in this article.

8. CONCLUSIONS AND FUTURE WORK

We have developed an unsupervised learning framework for extracting precise popular product attribute text fragments from product description pages originated from different Web sites. One characteristic of our framework is that the set of popular product attributes is unknown in advance, yet they can be extracted considering the interest of customers through an automatic identification of hidden concepts derived from a collection of customer reviews. Another characteristic of our framework is the capability of handling the vocabulary gap between the texts in Web product descriptions and texts in customer reviews or comments. We have conducted extensive experiments and comparison between existing approaches on a large number of Web pages from 13 different domains to demonstrate the effectiveness and robustness of our framework.

We intend to extend our framework in several directions. One possible direction is to develop an approach to exploiting a limited amount of training examples from users. Sometimes, users may have collected a few training examples about the attributes of the products in a domain. These training examples can be used to enhance the training

of our model. Another possible direction is to incorporate users' prior knowledge of a domain. Very often, users may have some prior knowledge about some common and important features of the products. Such prior knowledge can be helpful in deriving the popular features from customer reviews.

REFERENCES

Enrique Alfonseca, Marius Pasca, and Enrique Robledo-Arnuncio. 2010. Acquisition of instance attributes via labeled and related instances. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 58–65.

Lidong Bing, Wai Lam, and Yuan Gu. 2011. Towards a unified solution: Data record region detection and segmentation. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*. ACM, New York, NY, 1265–1274.

Lidong Bing, Wai Lam, and Tak-Lam Wong. 2013. Wikipedia entity expansion and attribute extraction from the web using semi-supervised learning. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM'13)*. ACM, New York, NY, USA, 567–576.

Lidong Bing, Tak-Lam Wong, and Wai Lam. 2012. Unsupervised extraction of popular product attributes from web sites. In *Proceedings of the 8th Asia Information Retrieval Societies Conference*. 437–446.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

Kenneth Bloom, Navendu Garg, and Shlomo Argamon. 2007. Extracting appraisal expressions. In *Proceedings of Human Language Technologies/North American Association of Computational Linguists*. Association for Computational Linguistics, Rochester, New York, 308–315.

D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. 2004. Block-based web search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 456–463.

Xiaowen Ding, Bing Liu, and Lei Zhang. 2009. Entity discovery and assignment for opinion mining applications. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 1125–1134.

J. L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychol. Bull.* 76, 5 (1971), 378–382.

Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text mining for product attribute extraction. *SIGKDD Explor. Newslett.* 8, 1 (2006), 41–48.

H. Guo, H. Zhu, Z. Guo, Z. Zhang, and Z. Su. 2009. Product feature categorization with multilevel latent semantic association. In *Proceedings of the 18th ACM International Conference on Information and Knowledge Management*. ACM, New York, NY, 1087–1096.

Minqing Hu and Bing Liu. 2004a. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 168–177.

Minqing Hu and Bing Liu. 2004b. Mining opinion features in customer reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence (AAAI'04)*. 755–760.

Nozomi Kobayashi, Kentaro Inui, Yuji Matsumoto, Kenji Tateishi, and Toshikazu Fukushima. 2004. Collecting evaluative expressions for opinion extraction. In *Proceedings of the International Joint Conference on Natural Language Processing*. 584–589.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning*. 282–289.

Xiao Li, Ye-Yi Wang, and Alex Acero. 2009. Extracting structured information from user queries with semi-supervised conditional random fields. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 572–579.

Bing Liu, Robert Grossman, and Yanhong Zhai. 2003. Mining data records in web pages. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*. ACM, New York, NY, USA, 601–606.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*. ACM, New York, NY, USA, 342–351.

Ping Luo, Fen Lin, Yuhong Xiong, Yong Zhao, and Zhongzhi Shi. 2009. Towards combining web classification and web information extraction: A case study. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 1235–1244.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, 339–346.

K. Probst, M. Krema R. Ghai, A. Fano, and Y. Liu. 2007. Semi-supervised learning of attribute-value pairs from product descriptions. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. 2838–2843.

Changqin Quan and Fuji Ren. 2014. Unsupervised product feature extraction for feature-oriented opinion determination. *Inf. Sci.* 272 (2014), 16–28.

A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. 2007. Hidden conditional random fields. *IEEE Trans. Pattern Anal. Mach. Intell.* 29(10) (2007), 1848–1853.

Xinying Song, Jing Liu, Yunbo Cao, Chin-Yew Lin, and Hsiao-Wuen Hon. 2010. Automatic extraction of web data records containing user-generated content. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, New York, NY, 39–48.

Y.-H. Sung and D. Jurafsky. 2009. Hidden conditional random fields for phone recognition. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 107–112.

Huifeng Tang, Songbo Tan, and Xueqi Cheng. 2009. A survey on sentiment detection of reviews. *Expert Syst. Appl.* 36 (September 2009), 10760–10773. Issue 7.

Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*. ACM, New York, NY, USA, 111–120.

Peter D. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, 417–424.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 783–792.

Tak-Lam Wong, Lidong Bing, and Wai Lam. 2011. Normalizing web product attributes and discovering domain ontology with minimal effort. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM'11)*. ACM, New York, NY, 805–814.

Tak-Lam Wong and W. Lam. 2007. Adapting web information extraction knowledge via mining site invariant and site depdent features. *ACM Trans. Internet Technol.* 7(1) (2007), Article 6.

Tak-Lam Wong, W. Lam, and T. S. Wong. 2008. An unsupervised framework for extracting and normalizing product attributes from multiple web sites. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 35–42.

Liheng Xu, Kang Liu, Siwei Lai, and Jun Zhao. 2014. Product feature mining: Semantic clues versus syntactic constituents. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22–27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*. 336–346.

Chunyu Yang, Yong Cao, Zaiqing Nie, Jie Zhou, and Ji-Rong Wen. 2010. Closing the loop in webpage understanding. *IEEE Trans. Knowledge Data Eng.* 22 (May 2010), 639–650. Issue 5.

Yanhong Zhai and Bing Liu. 2006. Structured data extraction from the web based on partial tree alignment. *IEEE Trans. Knowledge Data Eng.* 18(12) (2006), 1614–1628.

Lei Zhang, Bing Liu, Suk Hwan Lim, and Eamonn O'Brien-Strain. 2010. Extracting and ranking product features in opinion documents. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. 1462–1470.

Xin Wayne Zhao, Yanwei Guo, Yulan He, Han Jiang, Yuexin Wu, and Xiaoming Li. 2014. We know what you want to buy: A demographic-based system for product recommendation on microblogs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*. ACM, New York, NY, 1935–1944.

Shuyi Zheng, Ruihua Song, Ji-Rong Wen, and C. Lee Giles. 2009. Efficient record-level wrapper induction. In *Proceeding of the 18th ACM International Conference on Information and Knowledge Management*. ACM, New York, NY, 47–56.

J. Zhu, Z. Nie, B. Zhang, and J.-R. Wen. 2008. Dynamic hierarchical Markov random fields for integrated web data extraction. *J. Mach. Learn. Res.* (2008), 1583–1614.

Received November 2013; revised November 2015; accepted December 2015