Unsupervised Extraction of Popular Product Attributes from Web Sites *

Lidong Bing[1], Tak-Lam Wong[2], and Wai Lam[1]

[1] Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
{ldbing, wlam}@se.cuhk.edu.hk
[2] Department of Mathematics and Information Technology
The Hong Kong Institute of Education
tlwong@ied.edu.hk

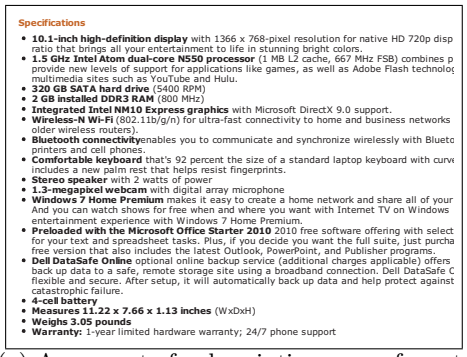
Abstract. We develop an unsupervised learning framework for extracting popular product attributes from different Web product description pages. Unlike existing systems which do not differentiate the popularity of the attributes, we propose a framework which is able not only to detect concerned popular features of a product from a collection of customer reviews, but also to map these popular features to the related product attributes, and at the same time to extract these attributes from description pages. To tackle the technical challenges, we develop a discriminative graphical model based on hidden Conditional Random Fields. We have conducted experiments on several product domains. The empirical results show that our framework is effective.

Keywords: Information Extraction, Conditional Random Fields

1 Introduction

For developing intelligent E-business systems, an important building block is to automatically extract attribute information from product description pages from different Web sites. For example, a product description page may contain a number of product attributes such as resolution and ISO of a digital camera in the digital camera domain. Existing automatic Web information extraction techniques including wrappers aim at extracting the product attributes from Web pages [1, 11, 17]. The product attributes of interest are normally specified by users, requiring a substantial amount of domain knowledge and manual effort. On the other hand, users usually are interested in a subset of the product attributes that are relevant to some features for making purchasing decision.

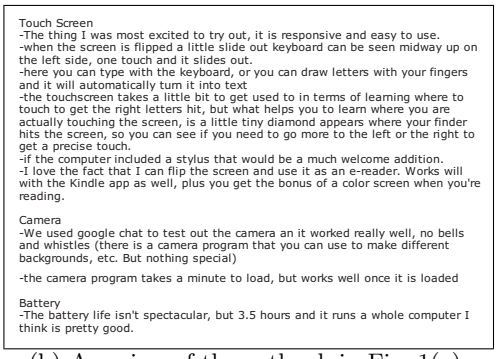
* The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: CUHK413510) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050476 and 2050522). This work is also affiliated with the CUHK MoE-Microsoft Key Laboratory of Human-centric Computing and Interface Technologies.



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 Bluetc printers and cell phones.
- **Comfortable keyboard** that's 92 percent the size of a standard laptop keyboard with curv 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 C 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

(a) 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.

(b) A review of the netbook in Fig. 1(a).

Fig. 1. Examples of product description page and user review.

For example, in the digital camera domain, users mainly consider the feature "picture quality", which is related to several product attributes including "resolution", "ISO", etc. Some other features such as "supported operating systems" may constitute tiny influence on users' decision making. Existing information extraction methods cannot automatically identify the product attributes that are of the users' interest. Moreover, these kinds of attributes are usually unknown in different domains. As a result, it raises the need for a method that can automatically identify the product attributes that are of users' interests and extract these attributes from Web pages.

We develop an unsupervised learning framework for extracting popular product attributes from different Web product description pages. Unlike existing systems which do not differentiate the popularity of the attributes, we propose a framework which is able not only to detect concerned popular features of a product from a collection of customer reviews, but also to map these popular features to the related product attributes, and at the same time to extract these attributes from description pages. We explain the rationale of our framework using the following example. Fig. 1(a) 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 attribute information of the netbook. However, not all of them are of interests to most customers and influence users' decision. We wish to extract those attributes which are important for customers to make decision. 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 Fig. 1(b) to automatically derive the popular features. Note that the concerned product from the Web page does not necessarily appear 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 features. For example, terms like "screen" and " color" are automatically identified to be re-

lated to the popular feature "display" of a netbook by analyzing their frequency and co-occurrence information in the customer reviews. Next these terms can help extract the text fragment "10.1-inch high-definition display ..." because it contains the terms "screen" and "color". Our framework can then reinforce that terms like "resolution" and "high-definition", which are contained in the text fragment are also related to the popular feature "display". These newly identified terms can be utilized to extract other attributes related to "display". On the other hand, some other attributes such as "keyboard" are not mentioned in most reviews. Hence, the text fragment "Comfortable keyboard ..." will not be extracted.

2 Problem Definition and Framework Overview

In a particular domain, let $\boldsymbol{A} = \{A_1, A_2, \ldots\}$ be the set of product attributes characterizing the products in this domain. For example, the set of product attributes of the netbook domain includes "screen", "multi-media", etc. 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 in W. 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 the text fragment $tok_{l,k}$ is related to the attribute A_j.

We denote $\boldsymbol{A_{POP}} \subseteq \boldsymbol{A}$ as the set of popular product attributes. Recall that $\boldsymbol{A_{POP}}$ is related to the popular features, namely, $\boldsymbol{C(R)}$, discovered from a collection of customer reviews, namely, $\boldsymbol{R}$, 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 in a domain and a set of customer reviews $\boldsymbol{R}$ in the same domain. The concerned product in the Web page W does not necessarily appear in R. We aim at automatically identifying all the possible text fragments $tok_{l,k}$ where $1 \leq l \leq k \leq N(W)$ 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 $\boldsymbol{A_{POP}}$ are automatically derived from $\boldsymbol{C(R)}$ beforehand and does not need to be pre-specified in advance.

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, W can be treated 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 of labels

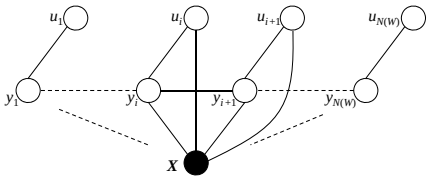


Fig. 2. The graphical model for popular attribute extraction. (Note that all u and y are connected to X in the model. Some obvious links are not shown for clarity.)

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 is $A_j \in \boldsymbol{A_{POP}}$, i.e. the type of popular attributes. Conditional Random Fields (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 due to several reasons. The first reason is that each token will be labeled by two kinds of labels simultaneously, whereas standard CRF considers only one kind of labels. 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 CRF cannot be employed. To tackle this problem, we have developed a graphical model based on hidden CRF. 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 derive $\boldsymbol{A_{POP}}$ from a collection of customer reviews $\boldsymbol{R}$. This component generates a set of derived documents from $\boldsymbol{R}$. We develop a method for selecting important terms for constructing the derived documents. Latent Dirichlet Allocation (LDA) is then employed to discover latent concepts, which essentially refer to the popular features of the products $\boldsymbol{C(R)}$, from the derived documents [2]. 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", etc., while another popular feature is more likely to generate the terms "camera", "speaker", etc. By making use of this term information, our graphical model can extract the text fragments related to the popular attributes.

3 Description of Our Framework

3.1 Our Model

Fig. 2 shows the graphical model capturing the inter-dependency among the essential elements in the extraction problem. Each node and edge of the graphical model represent a random variable and the dependence between two connected nodes. Recall that given a Web page, we can conduct some simple preprocessing by analyzing the DOM structure. The text content in the Web page can be

decomposed into a sequence of tokens $(tok_1, \ldots, tok_{N(W)})$. A random variable $\boldsymbol{X}$ refers to the observation from the sequence. For example, it can be 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 $\mathcal{Y}$ 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.

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:

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

where $f_k(e, y|_e, x)$ refers to the feature function related to x, the nodes ys connected by the edge $e \in \boldsymbol{E^Y}$. Referring to the text fragments "10.1-inch high-definition display ...", where tok_1 = "10.1-inch", tok_2 = "high-definition", ..., we may design a feature function $f_k(e, y|_e, x) = 1$ if $y_1 = B$, $y_2 = I$, x_1 contains a number, and $f_k(e, y|_e, x) = 0$ otherwise. $g_k(v, y|_v, x)$ refers to the feature function related to x, the node v represented by the vertex $v \in \boldsymbol{V}$; Similarly, we may design a feature function $g_k(v, y|_v, x) = 1$ if $y_i = I$ and x_i is the word "high-definition", and $g_k(v, y|_v, x) = 0$ otherwise. $h_k(e, y|_e, u|_e, x)$ refers to the feature function related to x, the nodes u and y connected by the edge $e \in \boldsymbol{E^U}$. For example, we may design a feature function $h_k(e, y|_e, u|_e, x) = 1$ if $y_i = I$, u_i refers to the popular product attribute "display" and x_i is the word "high-definition", and $h_k(e, y|_e, u|_e, x) = 0$ otherwise. 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.

3.2 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 [7] to describe our

method. We add the special label "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:

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

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

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

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

where $Z(x) = (\prod_{i=1}^{N(W)+1} M_i(y', y|x))_{\text{start,end}}$ is the normalization factor. Given the sequence of tokens and its observation, we can compute the optimal labeling of y and u using dynamic programming.

3.3 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:

$$\begin{aligned} \mathcal{L}_\theta &= \sum_{m=1}^{M} \log P(u^{(m)}|x^{(m)}; \theta) \\ &= \sum_{m=1}^{M} \log \sum_{y' \in y} P(y', u^{(m)}|x^{(m)}; \theta). \end{aligned} \tag{6}$$

Because of maximizing this log-likelihood function is intractable, we derive its lower bound according to Jensen's inequality and the concavity of the logarithm function. Then the efficient limited memory BFGS optimization algorithm is employed to compute the optimal parameter sets.

3.4 Hidden Concept Discovery

We observe that most customer reviews are organized in paragraphs as exemplified by the reviews in Fig. 1(b). To facilitate the discovery of high quality hidden concepts, we treat each paragraph as a processing document unit. For each review R, we first detect sentence boundaries in R using a sentence segmentator, R becomes a set of sentences denoted as $R = \{S_1, S_2, \ldots\}$. Let $S_i = (\eta_1^i, \eta_2^i, \ldots)$ denote a sequence of tokens in the sentence. Then linguistic parsing is invoked for each S_i to construct a parse tree, in which the constituents of S_i are organized in a hierarchical structure. We extract all the noun phrases located in the leaf nodes of the parse tree. Let the sequence of noun phrases of S_i be represented by $(N_{i1}, N_{i2}, \ldots)$. For each N_{ij}, we construct the context that is useful for latent topic discovery by considering the surrounding terms within a window size in S_i. Then we define ξ_{ij} as the derived content of N_{ij}, and it is composed of all terms in N_{ij} and the context terms. For example, consider a sentence S_i as (`It's`, `almost`, `soundless`, `on`, `the`, `low`, `setting`, `which`, `is`, `what`, `we`, `used`, `in`, `his`, `old`, `,`, `tiny`, `bedroom`), extracted from the review of an air purifier. The first noun phrase N_{i1} in this sentence is (`the`, `low`, `setting`). If the context window size is 2, the derived content ξ_{i1} for this noun phrase is (`soundless`, `on`, `the`, `low`, `setting`, `which`, `is`). We remove all stop words from ξ_{ij} and conduct lemmatization for the remaining terms. The derived content representation ξ_i of S_i is obtained by $\xi_i = \bigcup_j \xi_{ij}$.

The derived document Υ for R can be obtained by gathering the derived content of all sentences. Therefore $\Upsilon = (\xi_1, \xi_2, \ldots)$. A collection of derived documents for the review collection $\boldsymbol{R}$ can be obtained as above. We employ Latent Dirichlet Allocation (LDA) to discover the hidden concepts, which essentially refer to the popular features, for a domain.

4 Experimental Results

We have conducted the experiment to evaluate the performance of our framework with product description pages from over 20 different online retailer Web sites covering 7 different domains as depicted in Table 1. In addition, we have collected more than 500 customer reviews in each domain similar to the ones shown in Fig. 1(b) from retailer Web sites. These reviews were fed into the hidden concept discovery algorithm and the number of latent topics was set to 30 for each domain.

Two annotators were hired to identify the popular product attribute text fragments from the product description pages for evaluation purpose. The annotators first read through the reviews. Then they discussed and identified popular features as well as some sample popular product attributes for the domain. After that, such information is used to guide the annotation work of the corresponding domain. Text fragments corresponding to popular attributes were manually identified from product description pages. The manually extracted popular attribute text fragments were treated as the gold standard for evaluation. The

Table 1. The details of the data collected for the experiments.

Domain Label	Domain Name	# of products	# of description pages
D1	baby car seat	15	36
D2	carpet cleaner	12	24
D3	disc player	11	22
D4	GPS device	12	22
D5	netbook	10	30
D6	printer	10	21
D7	purifier	12	21

agreement of popular attribute text fragments between the two annotators was about 91%. The others were eliminated from the gold standard.

Since there are no existing methods that directly extract popular product attributes from Web pages and take into account customers' interest revealed in the collection of reviews of the same domain in an unsupervised manner, we implemented a comparative method based on integration of some existing methods. The first comparative method is called "VIPS-Bayes", which consists of two steps. For the first step, we first conduct unsupervised Web data extraction based on VIPS [3]. Since we have observed that almost all of the popular product attribute values (text fragments) are noun phrases, we apply the openNLP[3] package to conduct noun phrase extraction from the text in the product description blocks. The identified noun phrases 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. Note that each hidden concept is represented by a set of terms with probabilities. The probability refers how likely that a term is generated from a particular derived hidden concept. Next, each popular attribute value candidate is scored using Bayes theorem. In essence, the score refers to the conditional probability that the candidate comes from a particular derived hidden concept. Those candidates with scores greater than a certain threshold will be considered as popular attribute values. The threshold value is determined by a tuning process so that for each domain the best performance can be obtained.

Table 2 depicts the extraction performance of each domain and the average extraction performance among all domains. It can be observed that our approach achieves the best performance. The average F1-measure of our approach is 0.672, while the average F1-measure value of "VIPS-Bayes" is 0.547. In addition, the paired t-test (with $P < 0.001$) shows that the performance of our approach is significantly better. It illustrates that our approach can leverage the clues to make coherent decision in both product attribute extraction task and popular attribute classification task, leading to a better performance. 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

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

Table 2. The popular attribute extraction performance of our approach and the comparative method. P, R, and F1 refer to the precision, recall, and F1-measure respectively.

Domain	Our Approach			VIPS-Bayes		
	P	R	F1	P	R	F1
D1	0.681	0.769	0.713	0.523	0.778	0.612
D2	0.630	0.868	0.718	0.482	0.945	0.620
D3	0.548	0.872	0.665	0.388	0.836	0.519
D4	0.606	0.808	0.684	0.354	0.791	0.474
D5	0.776	0.813	0.789	0.722	0.814	0.757
D6	0.479	0.779	0.563	0.372	0.834	0.495
D7	0.589	0.611	0.571	0.558	0.317	0.350
Avg.	0.616	0.789	0.672	0.486	0.759	0.547

are utilized, our hidden CRF model can effectively extract the popular attribute text fragments from description pages.

5 Related Work

Some information extraction approaches for Web pages rely on wrappers which can be automatically constructed via wrapper induction. For example, Zhu et al. developed a model known as Dynamic Hierarchical Markov Random Fields which is derived from Hierarchical CRFs (HCRF) [17]. Zheng et al. proposed a method for extracting records and identifying the internal semantics at the same time [16]. Yang et al. developed a model combing HCRF and Semi-CRF that can leverage the Web page structure and handle free texts for information extraction [14]. Luo et al. studied the mutual dependencies between Web page classification and data extraction, and proposed a CRF-based method to tackle the problem [9]. Some common disadvantages of the above supervised methods are that human effort is needed to prepare training examples and the attributes to be extracted are pre-defined.

Some existing methods have been developed for information extraction of product attributes based on text mining. Ghani et al. proposed to employ a classification method for extracting attributes from product description texts [5]. Probst et al. proposed a semi-supervised algorithm to extract attribute value pairs from text description [11]. Their approach aims at handling free text descriptions by making use of natural language processing techniques. Hence, it cannot be applied to Web documents which are composed of a mix of HTML tags and free texts. The goal of extracting popular product attributes from product description Web pages is different from opinion mining or sentiment detection research as exemplified in [4, 6, 8, 10, 12, 13, 15]. These methods typically discover and extract all product attributes as well as opinions directly appeared in customer reviews. In contrast, our goal is to discover popular product attributes from description Web pages.

6 Conclusions

We have developed an unsupervised learning framework for extracting precise popular product attribute text fragments from description pages originated from different Web sites. 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.

References

1. Alfonseca, E., Pasca, M., Robledo-Arnuncio, E.: Acquisition of instance attributes via labeled and related instances. In: SIGIR. pp. 58–65 (2010)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. JMLR 3, 993–1022 (2003)
3. Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: Block-based web search. In: SIGIR. pp. 456–463 (2004)
4. Ding, X., Liu, B., Zhang, L.: Entity discovery and assignment for opinion mining applications. In: KDD. pp. 1125–1134 (2009)
5. Ghani, R., Probst, K., Liu, Y., Krema, M., Fano, A.: Text mining for product attribute extraction. SIGKDD Explorations 8(1), 41–48 (2006)
6. Kobayashi, N., Inui, K., Matsumoto, Y., Tateishi, K., Fukushima, T.: Collecting evaluative expressions for opinion extraction. In: IJCNLP. pp. 584–589 (2004)
7. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML. pp. 282–289 (2001)
8. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: WWW. pp. 342–351 (2005)
9. Luo, P., Lin, F., Xiong, Y., Zhao, Y., Shi, Z.: Towards combining web classification and web information extraction: a case study. In: KDD. pp. 1235–1244 (2009)
10. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: HLT/EMNLP. pp. 339–346 (2005)
11. Probst, K., R. Ghai, M.K., Fano, A., Liu, Y.: Semi-supervised learning of attribute-value pairs from product descriptions. In: IJCAI. pp. 2838–2843 (2007)
12. Tang, H., Tan, S., Cheng, X.: A survey on sentiment detection of reviews. Expert Systems with Applications 36(7), 10760–10773 (2009)
13. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: ACL. pp. 417–424 (2002)
14. Yang, C., Cao, Y., Nie, Z., Zhou, J., Wen, J.R.: Closing the loop in webpage understanding. TKDE 22(5), 639–650 (2010)
15. Zhang, L., Liu, B., Lim, S.H., O'Brien-Strain, E.: Extracting and ranking product features in opinion documents. In: Coling: Posters. pp. 1462–1470 (2010)
16. Zheng, S., Song, R., Wen, J.R., Giles, C.L.: Efficient record-level wrapper induction. In: CIKM. pp. 47–56 (2009)
17. Zhu, J., Nie, Z., Zhang, B., Wen, J.R.: Dynamic hierarchical markov random fields for integrated web data extraction. JMLR 9, 1583–1614 (2008)