Product Feature Categorization with Multilevel Latent Semantic Association

Honglei Guo Huijia Zhu Zhili Guo XiaoXun Zhang and Zhong Su
IBM China Research Lab
Beijing, P.R.China, 100193
{guohl, zhuhuij, guozhili, zhangxx, suzhong}@cn.ibm.com

ABSTRACT

In recent years, the number of freely available online reviews is increasing at a high speed. Aspect-based opinion mining technique has been employed to find out reviewers' opinions toward different product aspects. Such finer-grained opinion mining is valuable for the potential customers to make their purchase decisions. Product-feature extraction and categorization is very important for better mining aspect-oriented opinions. Since people usually use different words to describe the same aspect in the reviews, product-feature extraction and categorization becomes more challenging. Manually product-feature extraction and categorization is tedious and time consuming, and practically infeasible for the massive amount of products. In this paper, we propose an unsupervised product-feature categorization method with multilevel latent semantic association. After extracting product-features from the semi-structured reviews, we construct the first latent semantic association (LaSA) model to group words into a set of concepts according to their virtual context documents. It generates the latent semantic structure for each product-feature. The second LaSA model is constructed to categorize the product-features according to their latent semantic structures and context snippets in the reviews. Experimental results demonstrate that our method achieves better performance compared with the existing approaches. Moreover, the proposed method is language- and domain-independent.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural language processing—*text analysis*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*

General Terms

Algorithms, Experimentation

Keywords

opinion mining, product feature categorization, latent semantic association

CIKM'09, November 2–6, 2009, Hong Kong, China.
Copyright 2009 ACM 978-1-60558-512-3/09/11 ...$10.00.

1. INTRODUCTION

With the dramatic growth of web's popularity, the number of freely available online reviews is increasing at a high speed. Amounts of websites, blogs and forums today allow users to post reviews or comments for various products or services. For many applications, simply judging the overall sentiment orientation (i.e. positive or negative) of a review unit is not sufficient. Reviewers usually praise some aspects of the product and bemoan it in other aspects. It is important to find out reviewers' opinions toward different product-features instead of the overall opinion in those reviews. In recent years, some aspect-based opinion mining methods [8, 9, 11, 13, 17, 18, 19] have been presented to capture reviewers' opinions toward different product aspects. Such finer-grained opinion mining are valuable for the potential customers to make their purchase decisions. Product-feature extraction and categorization is always a bottleneck for aspect-based opinion mining in real applications. The quality of aspect-based opinion mining can be considerably enhanced by extracting and categorizing product-features correctly.

Product-feature extraction and categorization is a very challenging task since people usually use different words to describe the same aspect in the reviews. Some product specifications are provided by the merchants. However, customers and merchants may use different words to refer to the same aspect. For example, "*photo*", "*picture*" and "*image*" all refer to the same aspect in digital camera reviews. Moreover, some product-features (e.g. "*value*", "*style*") can not be found in the specifications. For non-product domains (e.g. *local service reviews*, *book stores*, etc.), finding features here is more difficult due to the lack of specifications. Manually product-feature extraction and categorization is tedious and time consuming, and practically infeasible for the massive amount of products. Thus, it raises the need for automatically extracting and categorizing product-features from customer reviews. The existing statistical methods use high-frequency noun phrases as product-features. They may produce too many non product-features and miss many low-frequency terms and their variations, and terms with only one word [8]. Although supervised methods [11] work better, they require much more human efforts for labeling training data.

Pros: ***LCD****, nice* ***touch screen****, longer* **battery life**
Cons: *Horrible* ***picture quality***
Review: The ***touch screen*** was the selling feature for me. The ***LCD touch screen*** is nice and large. This camera also has very impressive ***battery life***. However the ***picture quality*** is very grainy.

Figure 1: Semi-structured Review

In recent years, lots of web sites (e.g. *CNET.com* and *Epin-*

ions.com) support a semi-structured customer review format (see Figure 1). The reviewer is asked to briefly enumerate his/her concerned product-features and opinion words in ***Pros*** and ***Cons*** separately, and also write a detailed review. Such semi-structured reviews provide useful clues for product-feature extraction.

In this paper, we propose an unsupervised product-feature extraction and categorization method with multilevel latent semantic association. We first extract product-feature candidates from the semi-structured *Pros* and *Cons* parts, and further verify the valid product-features by checking their re-occurrence in the detailed reviews. Second, we employ a multilevel latent semantic association (LaSA) method to categorize the product-features. The first LaSA model is constructed to capture latent semantic association among words. It groups words into a set of concepts according to their virtual context documents in the reviews. This model generates the latent semantic structure for each product-feature. The second LaSA model categorizes all the product-features according to their latent semantic structures and context snippets in the review corpus. The intuition behind our method is that words in one concept set will have similar semantic features or latent semantic association, and share syntactic and semantic context in the corpus. They can be considered as behaving in the same way in the product-feature categorization. The proposed multi-level LaSA method categorizes product-features on a semantic level rather than by shallow lexical comparison. It can better approximate the true underlying semantic category distribution in the domain without any labeled samples. Experimental results show that our method achieves better performance compared with the existing approaches. The proposed method is language- and domain-independent.

The remainder of the paper is organized as follows. Section 2 introduces some related work. Section 3 illustrates how to extract product-features from semi-structured reviews. Section 4 presents latent semantic association method. Section 5 presents a product-feature categorization method with multilevel latent semantic association. The experimental results are discussed in Section 6. The conclusion is given in Section 7.

2. RELATED WORK

Product-feature extraction and categorization is a key task for opinion mining. Liu et al. [11] and Hu et al. [8, 9] identify product-features using association rule mining. Liu et al. [11] also present a supervised association rule mining method for detecting product-features in the semi-structured reviews. This method works better through manually labeling training data. Popescu and Etzioni [13] identify product-features using PMI-based feature assessor. Scffidi et al. [15] employ probability-based heuristics and baseline statistics of words in English to identify product-features. Symbolic approaches and statistical approaches are basically two techniques for terminology finding. Symbolic approaches [6, 5] focus on finding terms from noun phrases. They produce too many non product-features. Statistical approaches [16] exploit the co-occurrence frequency of words to predict terms. They miss many low frequency terms, variations and single-word terms. Although high-frequent features are the "hot" features that people are most interested in for a given product, There are some features that only a small number of people talked about. These low frequency ones, which are mentioned by some reviewers, can also be interesting to some customers. Thus, it is a big challenge to effectively extract these infrequent features in the reviews.

Liu et al. [11] group product-features by the synonym set in WordNet and the semi-automated tagging of reviews. Titov and McDonald [18, 19] employ multi-grain topic model and rating information to identify coherent aspects in the reviews. We present a multilevel latent semantic association method for product-feature categorization without using rating information.

Some work also examined the importance of such semi-structured review format [11, 22, 10, 2]. Brananvan et al [2]. leverage free keyphrases annotation (i.e. all the phrases in *Pros* and *Cons*) to infer document-level semantic properties. Their goal is to predict the semantic properties supported by a previously unseen document. They cluster all the keyphrases based on the lexical comparison and the distribution similarity. The lexical comparison is based on the cosine similarity among the phrase words while the distributional similarity is quantified in terms of the co-occurrence of keyphrases across review texts. Different from their method, we present a multilevel LaSA method for product-feature categorization in opinion mining. The first LaSA model captures lexical semantic association among words according to their virtual context documents. It generates the latent semantic structure for each product-feature. The second LaSA model categorizes all the product-features according to their latent semantic structures and context snippets in the corpus. In addition, Wong et al. [22] extract and normalize product attributes from the structured product descriptions provided by merchants in multiple web sites. However, merchants and customers may use different words to refer to the same feature. In this paper, we focus on extracting and categorizing product-features from customer reviews for opinion mining. Kaji et al. [10] also employ statistical method to build opinion lexicon from the layout structures.

3. EXTRACTING PRODUCT FEATURES FROM SEMI-STRUCTURED CUSTOMER REVIEWS

Product-feature extraction is very important for aspect-based opinion mining. However, few product-feature lists are available. In this section, we present an unsupervised method to extract product-features from the semi-structured reviews.

In recent years, some popular web sites (e.g. CNET.com and Epinions.com) support a semi-structured customer review format (see Figure 1 in Section 1). Semi-structured reviews provide useful clues for extracting product-features. Since reviewers often briefly enumerate their concerned product-features and opinions in *Pros* and *Cons*, we may obtain many typical product-feature candidates from these short sentence segments.

We extract product-features from the review corpus using the following three steps. All the reviews are tokenized and tagged part of speech automatically in the preprocess.

1. Extract product-feature candidates from *Pros* and *Cons*.

   • Split all the sentence segments in *Pros* and *Cons* into fragments using opinion words and stop words;

   • Add fragments ending with nouns to the product-feature candidate set.

2. Verify all the candidates by detecting their re-occurrence in the detailed review part D_f.

   • If a candidate f_i re-occurs in D_f, f_i is valid. Otherwise, if f_i's substrings ending with nouns re-occur in D_f, these substrings are valid candidates.

   • Add the valid candidates to the product-feature list.

3. Expand the product-feature list by recalling compound terms from the review corpus.

   • Automatically tag product-feature terms in the review corpus using the product-feature list.

- Merge the continuous tagged product-feature terms as compound terms.

Algorithm 1 illustrates our unsupervised product-feature (PF) extraction method in detail.

Algorithm 1: Product-Feature Extraction Method

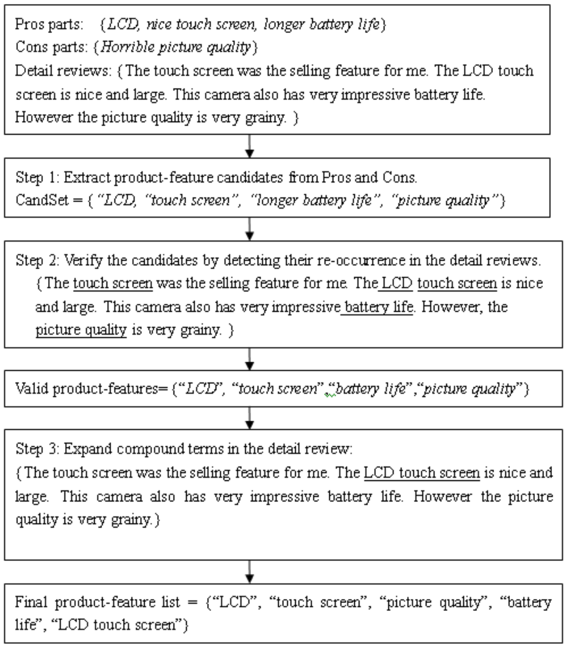
```
Inputs:
• $Corpus$: customer review corpus
Outputs:
• $FeatList$: the list of PFs.
Initialization:
• PF candidate set $CandSet = \emptyset$
Steps:
• foreach review $r_k \in Corpus$ do
    $D_s \leftarrow$ sentence segments from Pros and Cons in $r_k$;
    $D_f \leftarrow$ sentences from the detailed part in $r_k$;
    • Step 1: Extract PF candidates from $D_s$.
    foreach Sentence Segment $Seg_i \in D_s$ do
        Split $Seg_i$ into fragments using opinion words and stop words.
        Add all the fragments ending with nouns to $CandSet$.
    • Step 2: Verify PF candidates by checking their re-occurrences in $D_f$.
    foreach fragment $f_i \in CandSet$ do
        if $f_i$ re-occurs in $D_f$ then
            AddTo($f_i$, $FeatList$)
        else
            foreach substring $f_i'$ (ending with a noun) in $f_i$ do
                if $f_i'$ re-occurs in $D_f$ then
                    AddTo($f_i'$, $FeatList$)
• Step 3: Expand $FeatList$ by recalling the compound terms in $Corpus$
foreach Sentence $s_i \in Corpus$ do
    Tag PFs in $s_i$ using $FeatList$.
    Merge the continuous PFs in $s_i$ as compound terms.
Add all the compound terms into $FeatList$
```

Figure 2 shows an example of product-feature extraction. Given a review r_k, we first get the product-feature candidates {"*LCD*", "*touch screen*", "*longer battery life*", "*picture quality*"} from *Pros* and *Cons* parts by removing opinion words "*nice*" and "*horrible*". Second, we check their re-occurrence in the detail reviews and get the valid product-feature list {"*LCD*", "*touch screen*", "*battery life*","*picture quality*"}. Finally, a new compound term "*LCD touch screen*" is recalled by merging the continuous terms "*LCD*", and "*touch screen*". This proposed method can automatically extract product-features from the reviews without any labeled training data and hand-craft rules.

In order to filter opinion words in the sentence segments, we automatically build a list of opinion words from the corpus. Given a high frequency word x_i (e.g. the count of occurrence $\geq$ 10 in our experiments). Let $F_{pros}(x_i)$ and $F_{cons}(x_i)$ be the frequency of x_i in Pros and Cons corpus, respectively. We calculate the distribution difference between $F_{pros}(x_i)$ and $F_{cons}(x_i)$ using the similar method presented by Zagibalov et al. [23]. The distribution difference is calculated using Equation 1.

$$difference = \frac{2 \times |F_{pros}(x_i) - F_{cons}(x_i)|}{\sqrt{F^2_{pros}(x_i) + F^2_{cons}(x_i)}} \quad (1)$$

If *difference* $>$ 1, then the frequencies are not similar . Hence, x_i has enough distinguishing power. We consider x_i as an opinion word.



Pros parts: {*LCD, nice touch screen, longer battery life*}
Cons parts: {*Horrible picture quality*}
Detail reviews: {The touch screen was the selling feature for me. The LCD touch screen is nice and large. This camera also has very impressive battery life. However the picture quality is very grainy. }

Step 1: Extract product-feature candidates from Pros and Cons.
CandSet = {*"LCD, "touch screen", "longer battery life", "picture quality"*}

Step 2: Verify the candidates by detecting their re-occurrence in the detail reviews.
{The touch screen was the selling feature for me. The LCD touch screen is nice and large. This camera also has very impressive battery life. However, the picture quality is very grainy. }

Valid product-features= {*"LCD", "touch screen","battery life","picture quality"*}

Step 3: Expand compound terms in the detail review:
{The touch screen was the selling feature for me. The LCD touch screen is nice and large. This camera also has very impressive battery life. However the picture quality is very grainy.}

Final product-feature list = {"LCD", "touch screen", "picture quality", "battery life", "LCD touch screen"}

Figure 2: Example of product-feature extraction from customer reviews

4. LATENT SEMANTIC ASSOCIATION METHOD

The challenge in product-feature categorization is how to capture latent semantic association among product-features from the reviews. In this section, we present a latent semantic association (LaSA) model to find the latent semantic association structures of "words" according to their virtual context documents in the customer reviews.

4.1 Latent Semantic Association Model

Each product aspect often has various term representations. For example, "*photo*", "*picture*" and "*image*" all refer to the same aspect in the digital camera domain. It is a big challenge for product-feature categorization to effective capture latent semantic association among various product-features and their variants from the reviews. In this section, we present LaSA model to find the latent semantic association structures of "terms" according to their context snippets in the customer reviews.

Product-feature categorization focuses on grouping all the product-feature terms into semantic categories. We consider it as a concept group problem. Let X be a feature space to represent the observed word instances, and let Y be the set of semantic category labels. Let $p_s(x, y)$ and $p_t(x, y)$ be the predicted underlying category distribution and the true underlying target category distribution, respectively. In order to minimize the human efforts, we expect to $p_s(x, y)$ better approximate $p_t(x, y)$ without any labeled data.

Product-feature categorization based on lexical comparison usually is not comprehensive enough to capture the underlying semantic distribution of various product-features. Product-features in the same semantic category usually have various term representations. For example, "*screen*", "*display*", "*monitor*" and "*viewfinder*" all refer to the same aspect in the digital camera domain. Many product-feature terms in the same category are not similar on the lexical level. However, these terms often appear in the similar syntactic and semantic context. For instance, product-feature terms in the category "*appearance*" often occur around the indicates "*beautiful*", "*fashion*", "*popular*" etc. Such latent semantic association among words provides useful hints for capture the underlying semantic category distribution in the domain.

Hence, we present LaSA model $\theta_{s,t}$ to capture latent semantic association among words in the domain. $\theta_{s,t}$ is learned from the unlabeled domain data. Each instance is characterized by its co-occurred context distribution in the learning. Semantic association feature in $\theta_{s,t}$ is a hidden random variable that is inferred from data. Even though word instances do not have the lexical similarity, but are in similar context, they still might have relatively high probability in the same semantic concept set. Obviously, LaSA model can better capture the latent semantic association among words. It can enhance the estimate of the semantic category distribution $p_s(y|x;\theta_{s,t})$ to better approximate the real semantic category distribution $p_t(y|x;\theta_{s,t})$ in the domain.

In the product-feature categorization, we construct the first latent semantic association (LaSA) model to group words into a set of concepts according to their virtual context documents. Instances in the same concept set are considered as behaving in the same way for product-feature categorization. Thus, we employ the first LaSA model to generate the latent semantic structure for each product-feature. The second LaSA model is constructed to categorize the product-features according to their latent semantic structures and context snippets in the reviews. With the present multi-level LaSA method, we may better approximate the real semantic category distribution without any labeled data.

4.2 Learning LaSA Model from Virtual Context Documents

4.2.1 Virtual Context Document

In order to learn latent relationships among words from the unlabeled review corpus, each word candidate is characterized by a virtual context document as follows.

Given a word x_i, its virtual context document (denoted by vd_{x_i}) is composed of all the context units around x_i in the corpus, that is,

$$vd_{x_i} = \{F(x_i^{s_1}), ..., F(x_i^{s_k}), ..., F(x_i^{s_n})\}$$

where, $F(x_i^{s_k})$ denotes the context feature set of x_i in the kth sentence s_k, n is the total number of the sentences which contain x_i in the corpus.

Given the context window size {-t, t} (i.e. previous t words and next t words around x_i in s_k, t=3 in our experiments). $F(x_i^{s_k})$ usually consists of the following features.

1. Anchor unit $A_C^{x_i}$: the current focused word unit x_i.
2. Left opinion set $O_L^{x_i}$: two left adjacent adjective units $\{adj_{-1}, adj_{-2}\}$ around x_i, denoted by $\{O_L(adj_{-1}), O_L(adj_{-2})\}$.
3. Right opinion set $O_R^{x_i}$: two right adjacent adjective units $\{adj_1, adj_2\}$ around x_i, denoted by $\{O_R(adj_1), O_R(adj_2)\}$.
4. Left adjacent unit $A_L^{x_i}$: the nearest left adjacent unit x_{i-1} around x_i, denoted by $A_L(x_{i-1})$.
5. Right adjacent unit $A_R^{x_i}$: the nearest right adjacent unit x_{i+1} around x_i, denoted by $A_R(x_{i+1})$.
6. Left context set $C_L^{x_i}$: the other left adjacent units $\{x_{i-t}$, ..., x_{i-j}, ..., $x_{i-2}\}$ $(2 \leq j \leq t)$ around x_i, denoted by $\{C_L(x_{i-t})$, ..., $C_L(x_{i-j})$, ..., $C_L(x_{i-2})\}$.
7. Right context set $C_R^{x_i}$: the other right adjacent units $\{x_{i+2}$, ..., x_{i+j}, ..., $x_{i+t}\}$ $(2 \leq j \leq t$) around x_i, denoted by $\{C_R(x_{i+2}), ..., C_R(x_{i+j}), ..., C_R(x_{i+t})\}$.

For example, given x_i=*"screen"*, s_k=*"My new thinkpad is very good because its LCD screen is very large and nice"*. Let the context window size be {-3,3}. $F(screen)$ = {*screen*, O_L(*good*), O_L(*new*), O_R(*large*), O_R(*nice*), A_L(*LCD*), A_R(*is*), C_L(*its*), C_L (*because*), C_R(*very*), C_R(*large*)}.

vd_{x_i} actually describes the semantic and syntactic feature distribution of x_i in the domain. We construct the feature vector of x_i with all the observed context features in vd_{x_i}. Given the feature vector of vd_{x_i}, Vector(vd_{x_i})= $\{f_1^i, ..., f_j^i, ..., f_m^i\}$, f_j^i denotes the jth context feature related to x_i, m is the total number of features in vd_{x_i}. The weight of each context word f_j^i in vd_{x_i} is calculated by Mutual Information [4] between x_i and f_j^i.

$$Weight(f_j^i, x_i) = \log_2 \frac{P(f_j^i, x_i)}{P(f_j^i)P(x_i)} \tag{2}$$

where, $P(f_j^i, x_i)$ is the joint probability of x_i and f_j^i co-occurred in the corpus, $P(f_j^i)$ is the probability of f_j^i occurred in the corpus. $P(x_i)$ is the probability of x_i occurred in the corpus. The weight is normalized to non-negative in the model training.

4.2.2 Learning LaSA Model

LaSA model actually can be considered as a general probabilistic topic model. It can be learned from the unlabeled review corpus using the popular hidden topic models such as Latent Dirichlet Allocation (LDA) [1], probabilistic Latent Semantic Indexing (pLSI) [7].

Topic models are statistical models of text that posit a hidden space of topics in which the corpus is embedded [1]. LDA is a probabilistic model that can be used to model and discover underlying topic structures of documents. LDA assumes that there are K "topics", multinomial distributions over words, which describes a collection. Each document exhibits multiple topics, and each word in the document is associated with one of the topics. LDA imposes a Dirichlet distribution on the topic mixture weights corresponding to the documents in the corpus. The topics derived by LDA seem to possess semantic coherence. Those words with similar semantics are likely to occur in the same topic. Since the number of LDA model parameters depends only on the number of topic mixtures and vocabulary size, LDA is less prone to over-fitting and is capable of estimating the probability of unobserved test documents. LDA can be used to estimate the multinomial observations by unsupervised learning. LDA is already applied to enhance document representations in text classification [1] and information retrieval [21].

In the following, we illustrate how to construct LDA-style LaSA model θ on the virtual context documents. Algorithm 2 describes LaSA model training method in detail, where, Function $AddTo$ $(data, Set)$ denotes that $data$ is added to Set. Given a large-scale unlabeled review data set D_r, virtual context document for each word in the candidate word set $CandWordSet$ is extracted from D_r at first, and the weight of each context word in vd_{x_i} is computed using Mutual Information (see Equation 2 in Section 4.2.1). Then, LaSA model θ with Dirichlet distribution is generated on the virtual context document set $VDSet$ using the algorithm presented by Blei et al [1]. In our experiments, α=0.1, and the number of iterations is 1000.

LaSA model learns the posterior distribution to decompose words and their virtual context documents into topics. It extends the traditional bag-of-words topic models to context-dependence concept association model. It has potential use for concept grouping.

Algorithm 2: LaSA Model Training

```
Inputs:
• Customer review corpus: $D_r$;
• Candidate word set: $CandWordSet$;
Outputs:
• $LaSA$ model: $\theta$;
Initialization:
• Virtual context document set: $VDSet = \emptyset$;
Steps:
begin
    foreach $x_k \in CandWordSet$ do
        foreach sentence $S_i \in D_r$ do
            if $x_k \in S_i$ then
                $F(x_k^{S_i}) \longleftarrow \{x_k, O_L^{x_k}, O_R^{x_k}, A_L^{x_k}, A_R^{x_k}, C_L^{x_k}, C_R^{x_k}\}$;
                $AddTo(F(x_k^{S_i}), vd_{x_k})$;
        $AddTo(vd_{x_k}, VDSet)$;
    • Generate LaSA model $\theta$ with Dirichlet distribution on $VDSet$.
end
```

5. PRODUCT FEATURE CATEGORIZATION WITH MULTILEVEL LATENT SEMANTIC ASSOCIATION

Since people often use various terms to refer to the same feature, it is important to group product-features into semantic categories in opinion mining. In this section, we present a multilevel LaSA method to categorize the product-features. The first LaSA model is constructed for the words in the product-feature terms. It groups these words into a set of concepts according to their virtual context documents. This model generates the latent topic structure for each product-feature. The second LaSA model is constructed to categorize all the product-features according to their latent topic structures and context clues in the corpus.

5.1 Latent Topic Structure Generation for Product Features Using LaSA Model

Component words in the product-feature terms are important indicators for semantic categorization. In order to better categorize the product-features, we build LaSA model to generate the latent topic structures for product-features as follows.

LaSA model θ_W is constructed for the words in the product-features using Algorithm 2 (see Section 4.2.2). Virtual context documents of these words are constructed from the reviews using the method described in Section 4.2.1. LaSA model θ_W learns the posterior distribution to decompose these words and their virtual context documents into topics. Each word is assigned to a topic category by θ_W. Table 1 lists some content words from a random selection of 4 topics computed on the digital camera reviews. As shown, words in the same topic actually are grouped into broad concept sets. For example, topic 4, 8, 12 and 16 are related to the concepts *media*, *battery*, *photograph* and *screen*, respectively.

θ_W can effectively capture latent semantic association among the words. We employ θ_W to further generate the latent topic structures for product-features. The latent topic structure of a given product-feature pf_i is composed of the topic label sequence of all the words in pf_i, that is,

$$ts(pf_i)=\{ts(w_1), ..., ts(w_j), ..., ts(w_n)\}$$

Topic 4	Topic 8	Topic 12	Topic 16
card	AA	photo	viewfinder
slot	Alkaline	picture	monitor
stick	battery	shot	view
media	lithium	photograph	lcd
disk	charge	image	screen

Table 1: Samples from 4 randomly selected topics computed in digital camera domain

where, w_j denotes the jth word in pf_i. $ts(w_j)$ denotes w_j's topic assigned by θ_W. Table 2 shows the latent topic structures of some product-features generated by θ_W. By projecting the product-features into latent topic structures, we may capture more latent semantic similarity among those product-features with the same topic structure.

Product-feature Terms	Latent Topic Structure
memory card slot	*Topic20 Topic4 Topic4*
memory stick card	*Topic20 Topic4 Topic4*
memory stick media	*Topic20 Topic4 Topic4*
day photos	*Topic10 Topic12*
day pictures	*Topic10 Topic12*
day shots	*Topic10 Topic12*
daytime photos	*Topic10 Topic12*
indoor pictures	*Topic10 Topic12*

Table 2: Product-feature terms and their latent topic structures

5.2 Product Feature Categorization Based on Latent Topic Structure

The product-features with the similar latent topic structures usually have some semantic similarity. We categorize product-features according to their latent topic structures and context clues in the review corpus. In the following, we illustrate how to construct LaSA-based product-feature categorization model θ_P.

Given an unlabeled review corpus and the list of product-features, we first construct the LaSA model θ_W (see Section 5.1) to generate the latent topic structure for each product-feature. Second, the virtual context documents of all the latent topic structures are constructed from the review corpus. Finally, product-feature categorization model θ_P with Dirichlet distribution is generated from these virtual context documents using the algorithm presented by Blei et al [1]. θ_P learns the posterior distribution to decompose the latent topic structures of product-features and their context documents into topics. Hence, each product-feature is assigned to a topic category by θ_P. Algorithm 3 describes this method in detail, where, Function $TS(pf_i,\theta_W)$ denotes that θ_W generates the latent topic structure for the product-feature pf_i. Function $AddTo$ ($data$,Set) denotes that $data$ is added to Set.

In LaSA-based product-feature categorization model building, given a latent topic structure ts_j, its context document $CtxDoc_{ts_j}$ is composed of the context units of all the product-features with ts_j in the corpus, that is,

$$CtxDoc_{ts_j} = \sum_{1 \leq i \leq n} \sum_{1 \leq k \leq m} Ctx(pf_i, s_k)$$

where, pf_i denotes the ith product-feature with ts_j. n is the total number of the product-features with ts_j. $Ctx(pf_i, s_k)$ denotes the context units of pf_i in the kth sentence s_k. m is the total number of the sentences which contain pf_i in the corpus. In our experiments, $Ctx(pf_i, s_k)$ is composed of a bag of all the non-stop words in s_k. For example, given pf_i=*"LCD screen"*, s_k=*"Its LCD screen is very*

Algorithm 3: Product-feature Categorization Using Multi-level LaSA Method

```
Inputs:
• $D_r$: customer review corpus;
• $FeatList$: the list of product-features;
• $\theta_W$: $LaSA$ model of words in the product-features;
Outputs:
• Product-feature categorization model: $\theta_P$;
Initialization:
• Context document set: $CtxDocSet = \emptyset$;
• Latent topic structure set: $LTSet = \emptyset$;
Steps:
begin
    foreach $pf_i \in FeatList$ do
        $ts_j$=TS($pf_i$, $\theta_W$);
        if $ts_j$ is not in $LTSet$ then
            AddTo($ts_j$,$LTSet$);
        foreach Review $r_n \in D_r$ do
            foreach Sentence $s_k \in r_n$ do
                if $pf_i \in s_k$ then
                    $AddTo(Ctx(pf_i, s_k), CtxDoc_{ts_j})$;
    foreach $ts_i \in LTSet$ do
        $AddTo(CtxDoc_{ts_i}, CtxDocSet)$;
    • Generate product-feature categorization model $\theta_P$
    with Dirichlet distribution on $CtxDocSet$.
end
```

nice". Ctx("*LCD screen*", s_k) = { "*LCD*", "*screen*", "*nice*"}.

$CtxDoc_{ts_j}$ actually describes the latent semantic distribution of all the product-features with ts_j in the corpus. We construct the feature vector of ts_j with all the observed context features in $CtxDoc_{ts_j}$. Given the feature vector of $CtxDoc_{ts_j}$, $Vector$ ($CtxDoc_{ts_j}$) = {cf_1^j, ..., cf_i^j, ..., cf_M^j}, cf_i^j denotes the ith context feature related to ts_j, M is the total number of context features in $CtxDoc_{ts_j}$. The weight of cf_i^j is calculated as follows.

$$Weight(cf_i^j, ts_j) = tf \times \log_2 idf \qquad (3)$$

where, tf denotes the occurrence frequency of cf_i^j in $CtxDoc_{ts_j}$. idf denotes inverse document frequency which is the inverse of the percentage of the context documents in which cf_i^j occurs.

6. EXPERIMENTS

In this section, we evaluate the proposed approach using the customer reviews of three electronics product domains. Two data sets are English reviews in digital camera and laptop domains. One data set is Chinese reviews in cell phone domain. We analyze the performance of product-feature extraction and categorization in detail.

6.1 Data

Domain	Size (M)	Reviews	Language
Digital Camera	1.90	4,694	English
Laptop	0.74	2,348	English
Cell phone	5.07	8,181	Chinese
Total	7.71	15,233	–

Table 3: Review Data Sets

We build three domain-specific review data sets (see Table 3). The English digital camera and laptop data sets respectively consist of 4,694 customer reviews (totally 1.90M words) and 2,348 customer reviews (totally 0.74M words). The Chinese cell phone data set consists of 8,181 customer reviews (totally 5.07M Chinese characters). The English reviews are collected from several English product review web sites while the Chinese reviews are collected from a specialized cell phone review web site. Products in these sites have a large number of reviews. In the preprocess, these review documents are first cleaned to remove HTML tags. After that, a Maximum Entropy part-of-speech (POS) tagger is used to generate POS tags for English data. Meanwhile, Chinese data are segmented into words and tagged with POS using HMM.

6.2 Experimental Results

In the experiments, we extract and categorize the domain-specific product-features from the above three data sets. In the following, we will analyze these experimental results in detail.

6.2.1 Performance of Product Feature Extraction

We evaluate the quality of the product-feature extraction in this section. Three product-feature lists are respectively extracted from the above data sets.

We use the standard Precision (P) as the major performance metrics. To evaluation precision, we use each algorithm to generate list of outputs (product-features). All the output product-features are reviewed manually and then mark each as correct or incorrect. An output product-feature is correct if it refers to an attribute or component of the product in the domain. The quality is ensured by cross-validation checking.

Since noun phrase (NP) based statistical method (denoted as *NPStc*) is a popular unsupervised term extraction method, we compare our extraction method with it. *NPStc* method automatically extracts NP (i.e. two or more continuous nouns in a sentence) from the corpus, and outputs those high-frequency NPs as product-feature terms. In the following comparison experiments, *NPStc(Full)* method extracts all the NPs from all the full text in the data set, and outputs only high-frequency NPs (i.e. frequency $\geq$ 3) as product-features. *NPStc(Pros+Cons)* method extracts all the NPs from *Pros* and *Cons* parts in the data set, and outputs all the NPs as product-features.

Our product-feature extraction method consists of three steps, including *candidate extraction*, *verification* and *expansion*. In order to investigate the contribution of each step, we also compare our method (denoted as *Extract-Verify-Expand*) with the simplified *Extract-Verify* method. *Extract-Verify* method first extracts all the candidates from *Pros* and *Cons*, then verifies them in the detail reviews (see Step 1 and 2 described in Section 3). Finally, it outputs those verified valid candidates as product-features without any expansion.

Table 4 shows the accuracy of these methods in each domain. Experimental results show that *NPStc(Full)* and *NPStc(Pros+Cons)* produce too many non product-feature terms. Moreover, *NPStc(Full)* method misses many low-frequency ones. *Extract-Verify-Expand* outperforms *NPStc* and *Extract-Verify* in all the domains. Its accuracy is 0.8010, 0.7860 and 0.8269 in digital camera, laptop and cell phone domains, respectively. Furthermore, it recalls more correct product-feature terms than the other methods in each domain.

On the English digital camera data set, *Extract-Verify-Expand* yields an accuracy of 0.8010, and extracts 7,247 correct product-feature terms. Compared with *NPStc (Pros+Cons)* (P=0.7391), *Extract-Verify-Expand* significantly enhances the precision by 6.19 percent points. Especially, it recalls 5.08 times correct terms of

Methods	Digital Camera (English)		
	Output terms	Correct terms	Precision
NPStc(Full)	4,359	1,747	0.4008
NPStc(Pros+Cons)	1,690	1,249	0.7391
Extract-Verify	1,275	1,124	0.8816
Extract-Verify-Expand	9,048	7,247	0.8010
Methods	Laptops (English)		
	Output terms	Correct terms	Precision
NPStc(Full)	1,554	811	0.5219
NPStc(Pros+Cons)	1,621	616	0.3800
Extract-Verify	1,095	743	0.6785
Extract-Verify-Expand	4,065	3,195	0.7860
Methods	Cell phone (Chinese)		
	Output terms	Correct terms	Precision
NPStc(Full)	5,079	1,061	0.2089
NPStc(Pros+Cons)	363	259	0.7135
Extract-Verify	973	755	0.7760
Extract-Verify-Expand	5,512	4,558	0.8269

Table 4: Performance of product-feature extraction in the difference domains

NPStc (Pros+Cons) (1,249 correct terms). Compared with *NPStc(Full)* method (P=0.4008), *Extract-Verify-Expand* significantly enhances the precision by 40.02 percent points. Moreover, it recalls 4.15 times correct terms of *NPStc(Full)* method (1,747 correct terms). Experimental results on the English laptop review data set also show the similar trends.

On the Chinese cell phone data set, *Extract-Verify-Expand* method yields a precision of 0.8269, and extracts 4,558 correct product-feature terms. Compared with *NPStc(Pros+Cons)* (P=0.7135), it significantly enhances the precision by 11.34 percent points. Furthermore, it recalls 17.60 times correct terms of *NPStc(Pros+Cons)* method (259 correct terms). Compared with *NPStc(Full)* method (P=0.2089), *Extract-Verify-Expand* significantly improves the precision by 61.80 percent points. Meanwhile, it recalls 4.30 times correct terms of *NPStc(Full)* method (1,061 correct terms).

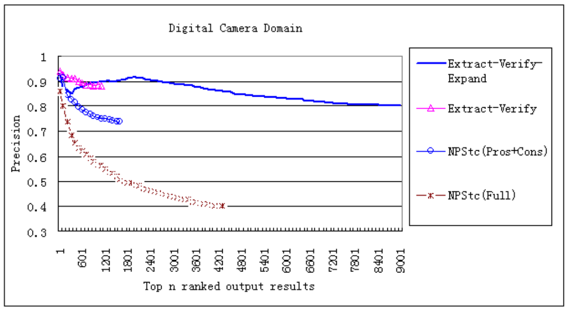


Figure 3: Precision curves of product-feature extraction in digital camera domain

Figure 3, 4 and 5 show precision curves of these methods in digital camera, laptop and cell phone domains. All the output candidates are ranked according to their frequency in the corpus. Given top n ranked outputs, *Extract-Verify-Expand* method achieves much better precision than the other methods at most of the cut-points. Moreover, its precision curve is more stable. On both digital camera and cell phone data sets, the precision of *Extract-Verify-Expand* method is above 80% at most of the cut-points while that one of the other methods ranges 80% $\sim$ 20% at most of the cut-points. On the laptop data set, the precision of *Extract-Verify-Expand* method is above 75% at most of the cut-points while that one of the other methods ranges 75% $\sim$ 40% at most of the cut-points. These experimental results show that *Extract-Verify-Expand* method effectively captures low-frequency terms. For example, it outputs 4,726 correct low-frequency terms (Frequency < 3) from the digital camera data set.

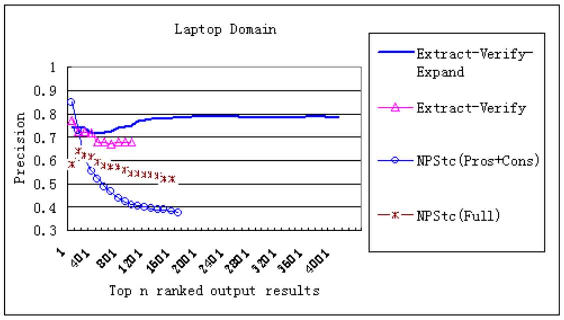


Figure 4: Precision curves of product-feature extraction in laptop domain

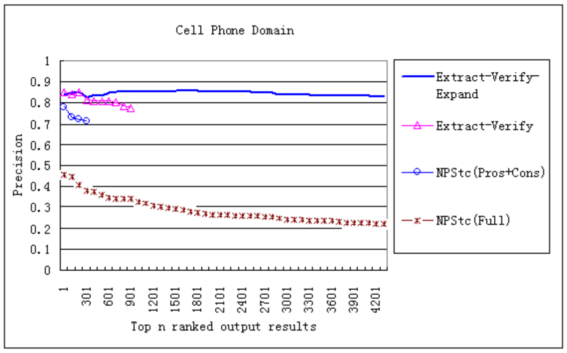


Figure 5: Precision curves of product-feature extraction in cell phone domain

All the experimental results show that our method achieves better performance. Our method effectively removes invalid candidates by reoccurring verification, and recalls more compound terms. It also captures more low-frequency product-features. Moreover, our method is language- and domain-independent.

Of course, our method still brings some noises in the product-feature extraction. The errors come from two aspects:1) Some candidates may not be the integrated phrases; 2) Not all fragments ending with nouns could be product-feature terms.

6.2.2 Evaluation of Product-Feature Category Construction

In this section, we evaluate the performance of product-feature categorization in the given three domains. In the experiments, the domain-specific product-feature lists extracted by *Extract-Verify-Expand* method (see Section 6.2.1) are further grouped into semantic categories, respectively.

With the limited of human efforts and time, we pre-constructed evaluation sets on several hot product-feature categories in each domain (see Table 5). All the product-features are checked manually. If a product-feature (PF) term satisfies the specification of product-feature categories, we give it the relevant label. The quality is ensured by cross-validation checking.

Domain (Language)	Category (Number of PF terms)	Total number of PF terms
Digital Camera (English)	Battery (246); Memory (483); Picture (143); Screen (259)	1,131
Laptop (English)	Battery (40); OS (81); Processor (119); Screen (123);	363
Cell phone (Chinese)	Appearance (417); Battery (97); Photograph (234); Screen (145);	893

Table 5: Categorization evaluation set

The performance of product-feature categorization is evaluated using Rand Index [14], a measure of cluster similarity [2, 3, 20].

$$\mathrm{Rand}(\mathrm{P}_1, \mathrm{P}_2) = \frac{2(a+b)}{n \times (n-1)} \tag{4}$$

In equation 4, P_1 and P_2 respectively represents the partition of an algorithm and manual labeling. The agreement of P_1 and P_2 is checked on their $n * (n-1)/2$ pairs of instances, where n is the size of data set D. For each two instances in D, P_1 and P_2 either assigns them to the same cluster or to different groups. Let a be the frequency where pairs belong to the same group of both partitions. Let b be the frequency where pairs belong to the different group of both partitions. Then Rand Index is calculated by the proportion of total agreement.

We used this measure to calculate the accuracy of product-feature categorization in the experiments. The parts of product-feature terms in the pre-constructed evaluation set are used to represent the data set D. Partition agreements between the pairs of any two terms in the parts and in the grouping results are checked automatically. This measure varies from zero to one. Higher scores are better.

We compared our multilevel LaSA method (denoted as *Multi-LaSA*, see Algorithm 3 in Section 5.2) with *k-means* clustering [12] and LDA-based categorization method (denoted as *LDA-based*). *k-means* is a standard clustering algorithm. In *k-means* clustering, each product-feature term is characterized by a bag of non-stop words in the sentences which contain this term. In *LDA-based* categorization method, the virtual context documents of all the product-features are constructed, and LDA-based categorization model for the product-features is learned from these virtual context documents using Blei's algorithm [1]. Finally, each product-feature is assigned to a category using this model.

In the experiments, the product-feature terms in each domain are categorized using these methods, respectively. Experimental results show that *Multi-LaSA* effectively groups the product-features into semantic categories. Figure 6, 7 and 8 list some categorization samples of *Multi-LaSA* in digital camera, laptop and cell phone domains, respectively. As shown, terms in each category are representative. Especially, although some terms have little similarity on the lexical level, they are also correctly grouped into the same category according to their semantic similarity. For example, on digital camera domain (see Figure 6), although terms "*screen*", "*LCD*", "*display*", "*viewfinder*" are not similar on the lexical level, they are correctly assigned to the same category "*Screen*". This shows that *Multi-LaSA* can effectively capture the latent semantic association among product-feature terms.

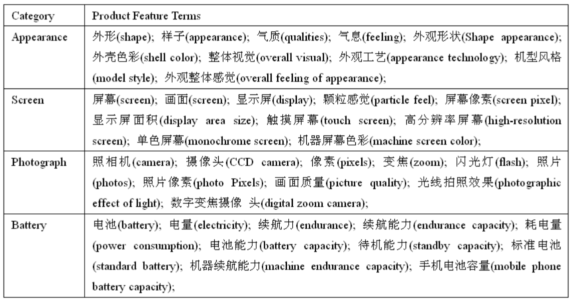
Selecting the right number of topics is also an important problem in the product-feature categorization. A range of 50 to 300 topics is typically used in the topic modeling literature. 50 topics are often

Category	Product-feature Terms
Battery	*battery life; Lithium battery; AA Alkaline; AA batteries; battery charger; battery capability; battery pack; battery adapter; AA Lithium batteries; rechargeable battery;*
Memory	*memory; flash card; flash memory; memory card; memory capacity; SD card; sd memory; camera flash; digital memory; CF memory card;*
Picture	*picture; auto picture; grainy image; grainy shots; cameras images; cameras photos; cameras pictures; color images; day shots; day picture;*
Screen	*screen; LCD; display; camera display; LCD panel; touch screen; LCD monitor; screen size; viewfinder; screen display;*

Figure 6: Product-feature categorization samples of *Multi-LaSA* in digital camera domain

Category	Product-feature Terms
Battery	*battery; power; battery power; Battery life; battery charger; battery warranty; machine batteries; Lithium cell; power management software; standard three-cell battery;*
OS	*Operating system; vista; windows xp machine; MAC OS; Apple; XP; MAC platform; operating systems windows; vista systems; windows xp home;*
Processor	*processor; CPU; core; core duo; core duo CPUs; solo processors; core speed; AMD chip; Macbook pros; Intel core duo CPU;*
Screen	*LCD; screen resolution; screen ratio; screen size; touch screen; LCD screen; wide screen resolution; glare; resolution; screen;*

Figure 7: Product-feature categorization samples of *Multi-LaSA* in laptop domain



Category	Product Feature Terms
Appearance	外形(shape); 样子(appearance); 气质(qualities); 气息(feeling); 外观形状(Shape appearance); 外壳色彩(shell color); 整体视觉(overall visual); 外观工艺(appearance technology); 机型风格(model style); 外观整体感觉(overall feeling of appearance);
Screen	屏幕(screen); 画面(screen); 显示屏(display); 颗粒感觉(particle feel); 屏幕像素(screen pixel); 显示屏面积(display area size); 触摸屏幕(touch screen); 高分辨率屏幕(high-resolution screen); 单色屏幕(monochrome screen); 机器屏幕色彩(machine screen color);
Photograph	照相机(camera); 摄像头(CCD camera); 像素(pixels); 变焦(zoom); 闪光灯(flash); 照片(photos); 照片像素(photo Pixels); 画面质量(picture quality); 光线拍照效果(photographic effect of light); 数字变焦摄像 头(digital zoom camera);
Battery	电池(battery); 电量(electricity); 续航力(endurance); 续航能力(endurance capacity); 耗电量(power consumption); 电池能力(battery capacity); 待机能力(standby capacity); 标准电池(standard battery); 机器续航能力(machine endurance capacity); 手机电池容量(mobile phone battery capacity);

Figure 8: Product-feature categorization samples of *Multi-LaSA* in cell phone domain

used for small collections and 300 for relatively large collections [21]. However, in the product-feature categorization, the number of topics might be set in a different range. It is confirmed here by our experiments with different values of K (10, 20, ...,100) on the three review data sets.

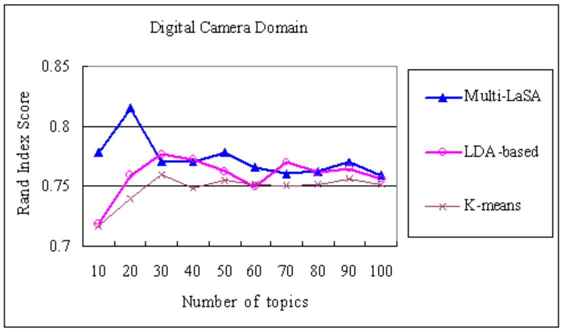


Figure 9: Product-feature categorization evaluation on digital camera domain with different number of topics (K)

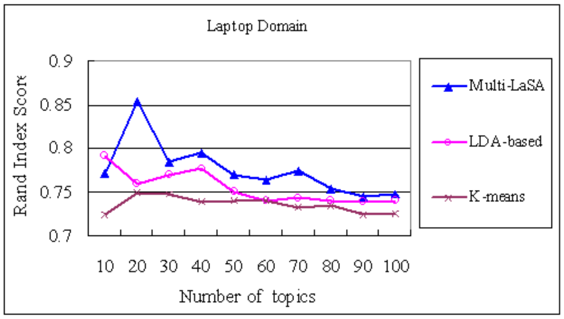


Figure 10: Product-feature categorization evaluation on laptop domain with different number of topics (K)

We evaluate the accuracy curves of these methods with different number of topics in each domain. Experimental results show that *Multi-LaSA* and *LDA-based* methods achieve better accuracy than *k-means* in each domain, as shown in Figure 9, 10 and 11. The major reason for significant performance enhancement is that *Multi-LaSA* and *LDA-based* methods effectively capture the latent semantic association among words from their virtual context documents. Moreover, *Multi-LaSA* also outperforms *LDA-based* method at the most of the operation points in each domain. Especially, at the number of topics K=20, the scores of *Multi-LaSA* method achieve 0.8154, 0.8537 and 0.8218 in digital camera, laptop and cell phone domains, respectively. Compared with *K-means*, *Multi-LaSA* significantly enhances the accuracy by 10.33%, 14.08% and 29.90%, respectively. Compared with *LDA-based* method, *Multi-LaSA* significantly improves the accuracy by 7.41%, 12.50% and 25.29%, respectively.

We perform t-tests on all the comparison experiments in all the three domains (see Figure 9, 10 and 11). On the 30 comparison experiments of *k-means* and *Multi-LaSA*, p-value is 4.93754E-07, which shows that the performance improvement is statistically significant. Meanwhile, p-value < 0.001 on the 30 comparison experiments of *LDA-based* method and *Multi-LaSA*, which shows that the enhancement is also statistically significant. Table 6 shows the performance comparison of these methods with different number of topics (K). The evaluation measure is average precision of each method on all the three domains with a predefined number-of-topics K. *Multi-LaSA* always obtains much better average precision than *k-means* and *LDA-based* methods at each predefined number-of-topics. Moreover, *Multi-LaSA* method gives the best average precision at K=20.

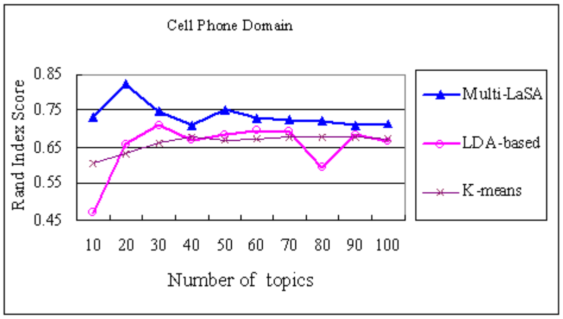


Figure 11: Product-feature categorization evaluation on cell phone domain with different number of topics (K)

Number of Topics(K)	Average precision			Δ P (%)	
	K-means	LDA-based	Multi-LaSA	(over k-means)	(over LDA-based)
10	0.6817	0.6604	0.7603	+11.53	+15.14
20	0.7067	0.7247	0.8303	+17.49	+14.58
30	0.7233	0.7519	0.7672	+6.07	+2.04
40	0.7207	0.7391	0.7590	+5.32	+2.70
50	0.7216	0.7320	0.7661	+6.17	+4.66
60	0.7209	0.7278	0.7536	+4.55	+3.54
70	0.7200	0.7349	0.7537	+4.69	+2.57
80	0.7210	0.6990	0.7461	+3.48	+6.74
90	0.7186	0.7297	0.7418	+3.24	+1.66
100	0.7171	0.7200	0.7407	+3.29	+2.88

Table 6: Comparison of k-means clustering, LDA-based method and Multi-LaSA with different number of topics (K). The evaluation measure is average precision of each method on all the three domains with a predefined number-of-topics (K). ΔP denotes the percentage change in performance (measured in average precision) of Multi-LaSA over k-means or LDA-based method.

All the above experimental results demonstrate that *Multi-LaSA* produces better categorizations than *LDA-based* method and *K-means*. The major reason for the significant enhancement is that *Multi-LaSA* better captures deeper latent semantic association among the product-features using latent topic structures. *Multi-LaSA* better approximates the underlying semantic distribution of the product-feature categories with a large-scale unlabeled customer review corpus.

Product-feature extraction and categorization is always a big challenge for opinion mining and other related real applications. This proposed method has been integrated into our existing opinion mining system. By grouping product-features into categories, our opinion mining system effectively provides non-trivial and more sound sentiment analysis and opinion summarization in several customer cases. We will further explore how to enhance the performance stability across domains in the real applications.

7. CONCLUSION

Fine-grained aspect-based opinion mining provides more useful information for users to make their purchase decision. It is very important to extract and categorize various product-features for opinion mining and other related real applications. However, few product-feature lists are available in practice. Although supervised extraction and categorization methods work better, they require lots of human efforts for labeling training data. In this paper, we propose an unsupervised product-feature extraction and categorization method with multilevel latent semantic association. We first extract the product-features from the semi-structured customer reviews. Second, we categorize the product-features using multilevel LaSA method. The first LaSA model captures latent semantic association among words in the product-features. It groups words into a set of concepts according to their virtual context documents. We employ this model to generate the latent semantic structure for each product-feature. The second LaSA model categorizes all the product-features according to their latent semantic structures and context snippets in the corpus. Experimental results show that our method achieves better performance than the existing approaches. Moreover, our method is language- and domain-independent.

8. REFERENCES

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(5):993–1022, 2003.

[2] S. Branavan, H. Chen, J. Eisenstein, and R. Barzilay. Learning document-level semantic properties from free-text annotations. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguisticsm (ACL'08)*, pages 263–271, 2008.

[3] C. Cardie and K. Wagstaff. Noun phrase coreference as clustering. In *Proceedings of the Joint Conf on Empirical Methods in NLP and Very Large Corpora*, pages 82–89, 1999.

[4] K. W. Church and P. Hanks. Word association norms, mutual information and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[5] P. Deane. A nonparametric method for extraction of candidate phrasal terms. In *Proceedings of 43th Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005.

[6] S. Evert and B. Krenn. Methods for the qualitative evaluation of lexical association measures. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics (ACL'01)*, 2001.

[7] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR99)*, 1999.

[8] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004)*, pages 761–769, 2004.

[9] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of Nineteeth National Conference on Artificial Intellgience (AAAI-2004)*, pages 755–760, 2004.

[10] N. Kaji and M. Kitsuregawa. Building lexicon for sentiment analysis from massive collection of html documents. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 1075–1083, 2007.

[11] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web (WWW'05)*, pages 1024–1025, 2005.

[12] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[13] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing(HLT-EMNLP -05)*, Vancouver, CA, 2005.

[14] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, December 1971.

[15] C. Scaffidi, K. Bierhoff, E. Chang, M. Felker, H. Ng, and C. Jin. Red opal: Product-feature scoring from reviews. In *Proceedings of the 8th ACM conference on Electronic commerce (EC'07)*, pages 182 – 191, 2007.

[16] P. Schone and D. Jurafsky. Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP'01)*, 2001.

[17] Q. Su, X. Xu, H. Guo, Z. Guo, X. Wu, X. Zhang, B. Swen, and Z. Su. Hidden sentiment association in chinese web opinion mining. In *Proceedings of the 17th international conference on World Wide Web (WWW'08)*, pages 959–968, 2008.

[18] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguisticsm (ACL'08)*, pages 308–316, 2008.

[19] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web (WWW'08)*, 2008.

[20] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, 2001.

[21] X. Wei and B. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR06)*, 2006.

[22] T.-L. Wong, W. Lam, and T.-S. Wong. An unsupervised framework for extracting and normalizing product attributes from multiple web sites. In *Proceedings of the 31st Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08)*, pages 35–41, 2008.

[23] T. Zagibalov and J. Carroll. Automatic seed word selection for unsupervised sentiment classification of chinese text. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1073–1080, 2008.