SpecLDA: Modeling Product Reviews and Specifications to Generate Augmented Specifications

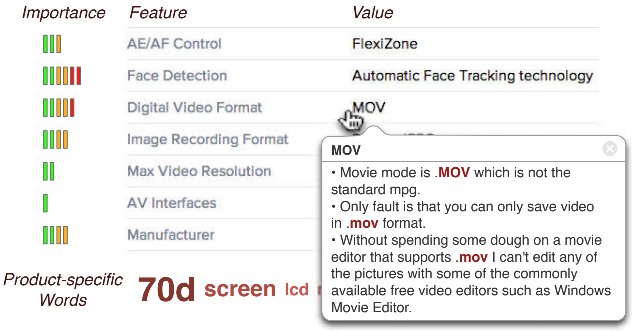
Dae Hoon Park* ChengXiang Zhai* Lifan Guo†

Abstract

Product specifications are often available for a product on E-commerce websites. However, novice customers often do not have enough knowledge to understand all features of a product, especially advanced features. In order to provide useful knowledge to the customers, we propose to automatically generate augmented product specifications, which contains relevant opinions for product feature values, feature importance, and product-specific words. Specifically, we propose a novel Specification Latent Dirichlet Allocation (SpecLDA) that can enable us to effectively model product reviews and specifications at the same time. It mines review texts relevant to a feature value in order to inform customers what other customers have said about the feature value in reviews of the same product and also different products. SpecLDA can also infer importance of each feature and infer which words are special for each product so that customers quickly understand products. Experiment results show that SpecLDA can effectively model product reviews with specifications. The model can be used for any text collections with specification (key-value) type prior knowledge.

1 Introduction

When people purchase a product from an online store, they are usually provided product-related information such as product description, product images, and user reviews. Often, product specifications are also provided to specify its features in an organized way, especially for high-technology products that consist of several electronic components. However, it is hard to understand what the contents of product specifications imply when the consumers are unfamiliar with them. For example, when novice consumers read a digital camera's specification, they probably do not have any idea what the value "TTL phase detection" of the feature "Auto Focus" means since they are not familiar with the feature value. Not only such consumers are unfamiliar with what the feature value is, but also they do not know how it truly means to them. In order to choose a product with the "right" value of a feature, they would like to hear direct experience from other consumers who own a product equipped with it, which may answer questions such as "is the feature value preferred by others?" and "is the feature considered important by others?" Finding out what people have said about a feature or a feature value across different products is a laborious task. The problem is worsen by the fact that many new high tech products have increasingly more new features. For example, a digital camera *Canon EOS 70D* has 79 features in CNET's product specifications page[1]. Thus, it is our goal to automatically augment product specifications through mining knowledge from product reviews and specifications across different products. Specifically, an augmented specification would show relevant review sentences, feature importance, and product-specific word list for each feature. An example of augmented specifications is shown in Figure 1.



Importance	Feature	Value
▮▮▮	AE/AF Control	FlexiZone
▮▮▮▮▮▮	Face Detection	Automatic Face Tracking technology
▮▮▮▮▮	Digital Video Format	MOV
▮▮▮▮	Image Recording Format	
▮▮	Max Video Resolution	
▮	AV Interfaces	
▮▮▮▮	Manufacturer	

Product-specific Words: 70d screen lcd

MOV

- Movie mode is .MOV which is not the standard mpg.
- Only fault is that you can only save video in .mov format.
- Without spending some dough on a movie editor that supports .mov I can't edit any of the pictures with some of the commonly available free video editors such as Windows Movie Editor.

Figure 1: An example of augmented specifications for a digital camera *Canon EOS 70D*.

The augmented specifications can be very useful for consumers. For example, the following sentence is retrieved from a review of a product with the feature-value pair ("Battery – Supported Battery", "Canon LP-E6 Li-ion rechargeable battery"). *The 60D uses the LP-E6 battery like the 7D, which is a nice feature as this battery can often last through a full day of shooting.* Through reading such user experience, consumers can learn about the feature value, and it will help them choose a proper product without reading all the reviews of products with

*Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA, {dpark34, czhai}@illinois.edu

†TCL Research America, 2870 Zanker Road, San Jose, CA 95134, USA, lifanguo@gmail.com

[1] http://www.cnet.com/products/canon-eos-70d/specs

that battery. In addition, the augmented specifications can provide how important the feature "Battery – Supported Battery" is to other customers and provide what characteristics are special for a certain product.

To the best of our knowledge, no previous work has addressed the novel problem of mining relevant opinions for a feature value in specifications across different products. Despite widespread existence of product specifications on the Web, only a few researchers studied specifications [31, 3, 27, 25, 19], and the goals of them are different from ours. This paper makes the following contributions:

1. We introduce and study a novel problem of mining product reviews (text data) and product specifications (structured data) jointly to discover relevant review sentences to each feature value in product specifications.

2. We propose a new probabilistic topic model, SpecLDA, to solve the proposed mining problem. SpecLDA can also infer each feature's importance and each product's specific theme. SpecLDA is general so that it can be applied to mine other kinds of text data and the companion structured data for alignment of values in structured data with sentences in text data.

3. We create a new data set for evaluating the new task and conduct experiments to show that the proposed SpecLDA outperforms a related state-of-the-art model for extracting relevant review sentences.

The potential impact of our model is significant. The augmented specifications can benefit product manufacturers as well as consumers. After learning what customers say about a feature or a feature value of products in a current market, they may focus on important features and develop components desired by consumers. In addition, since our suggested approach can be applied to any data set consisting of unstructured texts and specifications (key-value) data of entities, it can be employed in other problems.

2 Related Work

Finding opinions from a text data set have been widely studied, and several surveys summarized existing works [12, 13, 18]. Most of the studies performed research on product review [5, 8] or Weblog [17] data set since people leave rich opinions on them. In order to find the object of opinions and to mine opinions in a more effective way, aspect-based opinion mining and summarization [8, 21, 24] has been studied as a main stream in the field. Our work is related to the aspect-based opinion mining as we retrieve a user's review text on a particular product aspect. To find the aspects of a product, many studies [30, 17, 23] applied topic models [4, 7], which find latent topics from a text corpus. While most of the existing topic model-based approaches find latent topics without constraints on the topics, we utilize prior knowledge on the topics so that the topics and the specifications can match. Lu and Zhai [14] also used semi-supervised topic modeling with pre-defined topics, but their goal is to find opinions for a product aspect (feature), while we find opinions for each value of a product feature. Most existing studies in this line of research mine opinions on a product feature, either predefined or latent, but we mine opinions for a smaller grain of topic, a feature value.

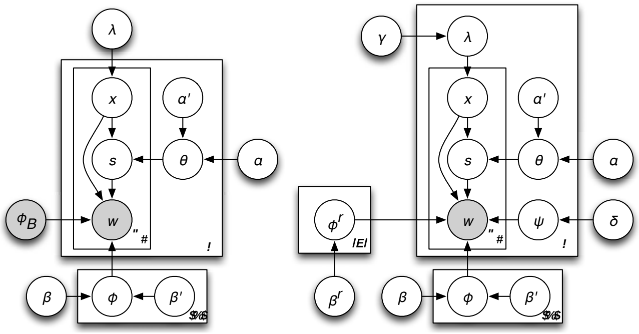
Although product specifications have been available in many e-commerce Web sites, only a limited number of studies employed them for product review analysis. Zhou and Chaovalit [31] established Ontology-Supported Polarity Mining (OSPM), which takes advantage of domain ontology database from IMDb[2], and their goal is sentiment classification on reviews. However, they studied only movie properties (features), not feature values. Bhattattacharya *et al.* [3] also used IMDb's structured data, but their goal is document categorization. Yu et al. [27] employed product specifications and reviews to build an aspect hierarchy, but they did not study feature values. Wang *et al.* [25] and Peñalver-Martínez *et al.* [19] also used product specifications to summarize product features, but neither of them studied feature values.

Modeling product reviews and specification simultaneously has been attempted in Duan *et al.* [6] with an extended PLSA model. However, the goal in Duan *et al.* [6] is to bridge the vocabulary gap in product search whereas our goal is to mine relevant review sentences to augment product specifications. Moreover, our proposed SpecLDA can better capture the feature structure in product specifications than the model used in Duan *et al.* [6] since their model does not consider hierarchy structure in specifications.

3 Problem Definition

The proposed new text mining problem is defined as follows. We are given M products $\boldsymbol{P} = \{P_1, ..., P_M\}$ with reviews $\boldsymbol{R}$, review sentences $\boldsymbol{T}$, and specifications $\boldsymbol{S}$. For each product p, there are specifications $\boldsymbol{S}_p$ and reviews $\boldsymbol{R}_p$ consisting of sentences $\boldsymbol{T}_p$. Specifications $\boldsymbol{S}_p$ of a product p is defined as $\boldsymbol{S}_p = \{s | s \in \boldsymbol{S}\ and\ s$ is part of $p\}$, where a specification s is a feature-value pair $(f,\ u)$, and $\boldsymbol{S}$ is a set of all possible feature-value pairs. Our goal is (1) to mine a set of sentences $\{t_1, ..., t_k\}$ for each specification (f, u), (2) predict importance for each feature f, and (3) mine a set of product-specific words for each product p. Please refer to Figure 1 as an example of output that we hope to generate.

[2] http://www.imdb.com



(a) DuanLDA (b) SpecLDA–

Figure 2: Graphical representation of DuanLDA and SpecLDA–.

This is a new problem that has not been addressed yet in previous work, and it can be regarded as a step toward an interesting new kind of mining problems involving both text data and the associated structured data. The problem is challenging for the following reason. The vocabulary used in specifications and reviews for a feature or a feature value may be different. Even if the vocabulary used in the reviews are the same as that in specifications, it is not known which words of the vocabulary people prefer to use to indicate a certain feature value. If we just use words of the given feature value, we may miss many of the relevant review sentences. Moreover, the data space is sparse since there may be many feature values for each of many features. Thus, stable estimation of statistics is necessary.

4 Methods

Generating augmented specifications using reviews is a new problem. There is no existing method that can be directly used to solve this problem. We propose to solve the problem by using a topic model for capturing the topics in review text, and impose a prior defined based on the feature values in the product specifications. This idea is essentially similar to the extended PLSA model proposed in Duan *et al.* [6] for product search. We thus first discuss how we can adapt this model to solve our problem.

4.1 DuanLDA In order to retrieve query-relevant products, Duan *et al.* [6] developed a probabilistic method that models product reviews and specifications. Their model can be regarded as a semi-supervised PLSA model with specifications as pre-defined topics and concatenated reviews as documents, which maximizes the following log-likelihood function of the whole data set:

$$l = \sum_{p \in \boldsymbol{P}} \sum_{w \in \boldsymbol{V}} c(w, r_p) \log \left[\lambda p(w|\boldsymbol{\phi}_B) + (1-\lambda) \sum_{s \in S} p(w|s)p(s|p)\right] \tag{4.1}$$

where $c(w, r_p)$ is a word count in concatenated reviews r_p for p, and $\boldsymbol{\phi}_B$ is a background language model. $\boldsymbol{\lambda}$ is a parameter for choosing either $\boldsymbol{\phi}_B$ or the predefined topics $\boldsymbol{S}_p$, and $\boldsymbol{V}$ is a vocabulary set for the corpus.

As discussed in [4, 28], LDA has several advantages over PLSA, so we convert their model to LDA version and call it DuanLDA. The graphical representation of DuanLDA is shown in Figure 2a. There are M product documents, where each document is a concatenated review for p, and there are N_p words for each document. s is a specification (a feature-value pair) topic, and there are $|\boldsymbol{S}|$ possible specifications. The generative story for each word is as following. When an author writes a review word $w_{p,i}$ at ith position of product document p, the author chooses a background topic or specification topics according to switch $x_{p,i}$, which is determined by a parameter λ. If the background topic is chosen, $w_{p,i}$ is drawn from background language model $\boldsymbol{\phi}_B$; otherwise, a specification $s_{p,i}$ is chosen according to $\boldsymbol{\theta}_p$, and $w_{p,i}$ is chosen according to $\boldsymbol{\phi}_{s_{p,i}}$. To incorporate pre-defined topics into LDA, product-specific topic distributions $\boldsymbol{\alpha}'_p$ and topic-specific word distributions $\boldsymbol{\beta}'_z$ are used to draw $\boldsymbol{\theta}_p$ and $\boldsymbol{\phi}_{s_{p,i}}$. Specifically, $\boldsymbol{\theta}_p$ is drawn from Dirichlet($\alpha\boldsymbol{\alpha}'_p$) and $\boldsymbol{\phi}_{s_{p,i}}$ is drawn from Dirichlet($\beta\boldsymbol{\beta}'_{s_{p,i}}$), and how to generate those prior distributions is explained later in this section.

The document language model for DuanLDA is defined by

$$p_{lda}(w|p, \lambda, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}) = \lambda p(w|\boldsymbol{\phi}_B) + (1-\lambda) \sum_{s=1}^{|\boldsymbol{S}|} p(w|s, \hat{\boldsymbol{\phi}})p(s|p, \hat{\boldsymbol{\theta}}) \tag{4.2}$$

The probability of $x_{p,i} = 0$, which chooses background language model, is determined by λ and the corpus $\boldsymbol{W}$, which is

$$p(x_{p,i} = 0|\boldsymbol{W}, \lambda) \propto \lambda p(w_{p,i}|\boldsymbol{\phi}_B) \propto \lambda \frac{N_{w_{p,i}}}{\sum_{w' \in \boldsymbol{V}} N_{w'}} \tag{4.3}$$

where N with superscript and/or subscript means the number of words satisfying the superscript and subscript conditions. Collapsed Gibbs sampling formula to choose $s_{p,i}$ when $x_{p,i} = 1$ is defined as

$$\begin{aligned} &p(x_{p,i} = 1, s_{p,i} = z|w_{p,i}, \boldsymbol{W}_{\setminus p,i}, \boldsymbol{S}_{\setminus p,i}, \lambda, \alpha, \boldsymbol{\alpha}'_p, \beta, \boldsymbol{\beta}'_{s_{p,i}}) \\ &\propto (1-\lambda)p(w_{p,i}|s_{p,i} = z, \boldsymbol{W}_{\setminus p,i}, \boldsymbol{S}_{\setminus p,i}, \beta, \boldsymbol{\beta}'_z)p(s_{p,i} = z|\boldsymbol{S}_{\setminus p,i}, \alpha, \boldsymbol{\alpha}'_p) \\ &\propto (1-\lambda) \frac{N^{\setminus p,i}_{w_{p,i}|z} + |\boldsymbol{V}|\beta\beta'_{z,w_{p,i}}}{N^{\setminus p,i}_{z} + |\boldsymbol{V}|\beta} \frac{N^{\setminus p,i}_{z|p} + K\alpha\alpha'_{p,z}}{N^{\setminus p,i}_{x=1|p} + K\alpha} \end{aligned} \tag{4.4}$$

where $\setminus p, i$ exclude a word at ith position of p.

Prior Generation In order to automatically generate priors $\boldsymbol{\beta}'_z$ for each topic z, we take an approach similar to that in [6]. However, the resulting distribution from their prior generation is quite even, so it does not distinguish important words from unimportant words well. Thus, we assume the prior words follows

Zipf's law distribution and adjust $p(w|f)$ according to it. Specifically, from the prior $p(w|f)$ obtained in [6], we define new word prior $p'(w|f)$ as

$$p'(w|f) = \begin{cases} \frac{p(w|f)}{\sum_{w \in \boldsymbol{V}(f)} p(w|f)} \sum_{i=1}^{|\boldsymbol{V}(f) \cap \boldsymbol{V}|} Zipf(i) & \text{if } w \in \boldsymbol{V}(f) \\ Zipf(rank_f(w) + |\boldsymbol{V}(f) \cap \boldsymbol{V}|) & \text{otherwise} \end{cases} \tag{4.5}$$

where $\boldsymbol{V}(f)$ is a vocabulary in f, $\boldsymbol{V}$ is a vocabulary in the review corpus, $rank_f(w)$ is w's rank in $p(w|f)$ excluding words in $\boldsymbol{V}(f)$, and Zipf's law distribution function $Zipf(i)$ is defined as $Zipf(i) = \frac{1/i^s}{\sum_{n=1}^{|\boldsymbol{V}|} 1/n^s}$, where s is a parameter characterizing the distribution. Basically, $p'(w|f)$ keeps the rankings in $p(w|f)$ but substitutes probabilities of non-feature words with Zipf's probability. $p(w|f)$ of feature words are redistributed to $p'(w|f)$ having sum of them equal to Zipf's probability sum for first $|\boldsymbol{V}(f) \cap \boldsymbol{V}|$ words. Then, we assign $\beta'_{s,w} = p'(w|f)$ for s whose feature is f. Also, we generate document-topic prior $\boldsymbol{\alpha}'$ based on specifications in each product; if a feature-value pair s is not present in a product p, we assign zero to $\alpha'_{p,s}$, and otherwise, we assign $\alpha'_{p,s}$ a probability, which is uniform among all present feature-value pairs.

4.2 SpecLDA: A topic model for joint mining of product reviews and specifications

DuanLDA has several deficiencies: (1) it considers only specification topics, (2) the prior amount ($|\boldsymbol{V}|\beta$) is uniform for all topics, (3) it does not fully take advantage of the specifications structure. We develop SpecLDA– (Figure 2b) that improves the points (1) and (2) and mines product-specific words. Then, we we further address point (3) and propose SpecLDA.

4.2.1 SpecLDA–

Review Topics Product reviews may have topics that are not in specifications; for example, value, design, or ease of use is not listed in specifications, but they may be mentioned in reviews. DuanLDA expects them to be captured by background language model, but it assumes that every product document has the same proportion of background topic, which may not be true. We thus remove background topic and add $|\boldsymbol{E}|$ review topics, resulting in all topics $\{s_1, \ldots, s_{|\boldsymbol{S}|}, s_{|\boldsymbol{S}|+1}, \ldots, s_{|\boldsymbol{S}|+|\boldsymbol{E}|}\}$, similar to those in [14]. Now, $\boldsymbol{\alpha}'_p$ contains uniform prior among present specifications and review topics, and zero for absent specifications. If the drawn topic $s_{p,i}$ belongs to specifications, it works the same as DuanLDA does. On the other hand, if $s_{p,i}$ belongs to review topics ($\boldsymbol{E}$), the word $w_{p,i}$ is drawn from $\boldsymbol{\phi}^r_{s_{p,i}}$, which is drawn from Dirichlet distribution with a symmetric prior β^r.

Prior Regularization Each specification topic s has its estimated topic size N_s. If the topic size N_s is relatively too small or too big compared to amount of prior, $|\boldsymbol{V}|\beta$, the topic s will rely too much or too little on the prior $\boldsymbol{\beta}'_s$. If a topic relies too much on prior, then the topic will just follow the word distribution of $\boldsymbol{\beta}'_s$, and if a topic relies too little on prior, it is likely to bear other themes that are unrelated to prior so that the topic is corrupted. Therefore, we need to regularize the prior size according to topic sizes, which is done similarly as in [22, 14]. We initialize prior size coefficient β with a big number, and we introduce prior size controllers $\{\eta_1, \ldots, \eta_{|\boldsymbol{S}|}\}$, each of which repeatedly decays by decay factor ζ if the topic size is too little.

Please note that we do not explicitly insert regularization variables to the Gibbs sampling formulas for SpecLDA for simplicity.

Product-specific Topics We add a product-specific topic $\boldsymbol{\psi}_p$ for each product p in order to capture how p is different from other products in reviews. In other words, if a word is closer to a product-specific topic than to any other topics, it is likely to be assigned with the product-specific topic. When a review author writes a word $w_{p,i}$ for a product p, the author first chooses between product-specific topic and specification topics according to $\boldsymbol{\lambda}_p$, which is drawn from Beta distribution with a symmetric prior γ. If the product-specific topic is chosen ($x_{p,i} = 0$), $w_{p,i}$ is drawn from $\boldsymbol{\psi}_p$, which is drawn from Dirichlet distribution with a symmetric δ.

4.2.2 SpecLDA

Feature-value pairs with the same feature share the feature information, but DuanLDA models them individually. We suggest to form a hierarchy among specifications, where we separate feature topics and value topics and share the feature topics for feature-value pairs with the same feature. By forming such hierarchy, we can expect SpecLDA to have more reliable topic estimation since it shares the feature topics and connects feature-value pairs who belong to the same feature. In addition, we can capture a user's preference on choosing feature-related or value-related words to indicate a feature-value pair. Another deficiency in Duan *et al.*'s model is that it imports priors only from feature words, not from value words. However, we believe that value word priors are also important, so we employ them as well as feature word priors. Value word priors are generated in the same way as feature word priors are.

The graphical representation of SpecLDA is depicted in Figure 3. For each feature f of all possible features $\boldsymbol{F}$, there are possible values $\boldsymbol{U}^f$. To separate a feature from feature values, a feature variable f is separated from the value variable u^f, which is a possible value for f. Also, the feature value topics $\boldsymbol{\omega}$ is introduced to separate them from feature topics $\boldsymbol{\phi}$. The generative story is as following. Choosing a word for product-specific is the same as in SpecLDA– . If a product feature is chosen by $x_{p,i}$, the author chooses a feature $f_{p,i}$ from the possible feature set $\{f_1, \ldots, f_{|\boldsymbol{F}|}, f_{|\boldsymbol{F}|+1}, \ldots, f_{|\boldsymbol{F}|+|\boldsymbol{E}|}\}$, which is a set of fea-

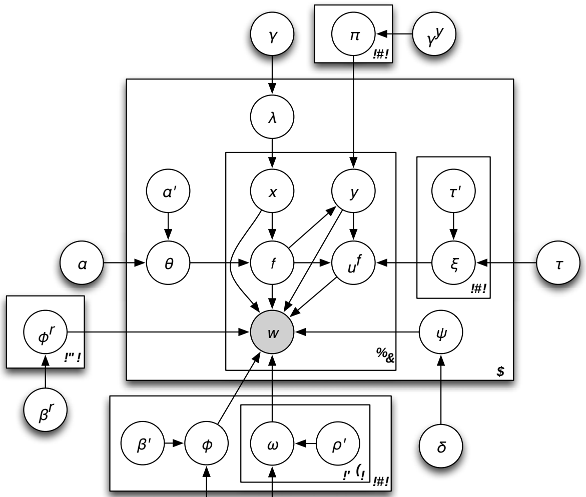


Figure 3: SpecLDA

ture and review topics, according to $\boldsymbol{\theta}_p$, which is drawn from Dirichlet distribution with $\alpha\boldsymbol{\alpha}'_p$. If $f_{p,i}$ belongs to review features, $w_{p,i}$ is drawn from multinomial distribution $\boldsymbol{\phi}^r_{f_{p,i}}$, which is drawn from Dirichlet distribution with a symmetric prior β^r. If the chosen feature $f_{p,i}$ belongs to specifications features, the author again chooses to write a feature or a value word using switch $y_{p,i}$ according to $\boldsymbol{\pi}_{f_{p,i}}$, which is drawn from beta distribution with a symmetric prior γ^y. If the author chooses to write a feature word, $w_{p,i}$ is chosen according to $\boldsymbol{\phi}_{f_{p,i}}$, which is drawn from Dirichlet distribution with $\beta\boldsymbol{\beta}'_{f_{p,i}}$. Otherwise, the author further chooses value $u^f_{p,i}$ for $f_{p,i}$ according to $\boldsymbol{\xi}_{p,f_{p,i}}$, which is drawn from Dirichlet distribution with $\tau\boldsymbol{\tau}_{p,f_{p,i}}$. With the chosen feature value $u^f_{p,i}$, the author chooses a word according to $\boldsymbol{\omega}_{f_{p,i},u^f_{p,i}}$, which is drawn from Dirichlet distribution with $\rho\boldsymbol{\rho}'_{f_{p,i},u^f_{p,i}}$. This process is repeated for all review words of all products.

The document language model of SpecLDA is thus:

(4.6)
$$p_{lda}(w|p, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\phi}}^r, \hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\psi}})$$
$$= p(x=0|\hat{\boldsymbol{\lambda}}_p)p(w|\hat{\boldsymbol{\psi}}_p) + p(x=1|\hat{\boldsymbol{\lambda}}_p)\big[\sum_{f\in\boldsymbol{E}} p(w|\hat{\boldsymbol{\phi}}^r_f)p(f|\hat{\boldsymbol{\theta}}_p)$$
$$+ \sum_{f\in\boldsymbol{F}_p} p(f|\hat{\boldsymbol{\theta}}_p)p(w|f, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\omega}})\big]$$

where

$$\begin{aligned} p(w|f, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\omega}}) &= p(y=0|\hat{\boldsymbol{\pi}}_f)p(w|\hat{\boldsymbol{\phi}}_f) \\ &+ p(y=1|\hat{\boldsymbol{\pi}}_f)\sum_{u\in\boldsymbol{U}^f} p(u|\hat{\boldsymbol{\xi}}_f)p(w|\hat{\boldsymbol{\omega}}_{f,u}) \end{aligned} \tag{4.7}$$

and the Gibbs sampling formula for learning when product-specific topic is used $(x = 0)$ is

(4.8)
$$p(x_{p,i}=0|w_{p,i}, \boldsymbol{W}_{\backslash p,i}, \boldsymbol{X}_{\backslash p,i}, \gamma, \delta)$$
$$\propto p(x_{p,i}=0|\boldsymbol{X}_{\backslash p,i}, \gamma)p(w_{p,i}|\boldsymbol{X}_{\backslash p,i}, \boldsymbol{W}_{\backslash p,i}, \delta)$$
$$\propto \frac{N^{\backslash p,i}_{x=0|p} + \gamma}{N_p - 1 + 2\gamma} \frac{N^{\backslash p,i}_{w_{p,i}|x=0} + \delta}{N^{\backslash p,i}_{x=0} + |\boldsymbol{V}|\delta}$$

To learn when we choose a review topic or a feature topic f, the formula is defined as

(4.9)
$$p(x_{p,i}=1, f_{p,i}=z, y_{p,i}=0|w_{p,i}, \boldsymbol{W}_{\backslash p,i}, \boldsymbol{X}_{\backslash p,i}, \boldsymbol{F}_{\backslash p,i}, \boldsymbol{E}_{\backslash p,i}, \boldsymbol{Y}_{\backslash p,i}, \Omega)$$
$$\propto p(x_{p,i}=1|\boldsymbol{X}_{\backslash p,i}, \Omega)p(f_{p,i}=z|\boldsymbol{F}_{\backslash p,i}, \boldsymbol{E}_{\backslash p,i}, \Omega)$$
$$p(y_{p,i}=0|z, \boldsymbol{Y}_{\backslash p,i}, \boldsymbol{F}_{\backslash p,i}, \boldsymbol{E}_{\backslash p,i}, \Omega)p(w_{p,i}|z, \boldsymbol{W}_{\backslash p,i}\boldsymbol{F}_{\backslash p,i}, \boldsymbol{E}_{\backslash p,i}, \boldsymbol{Y}_{\backslash p,i}, \Omega)$$
$$\propto \begin{cases} \frac{N^{\backslash p,i}_{x=1|p}+\gamma}{N_p-1+2\gamma} \times \frac{N^{\backslash p,i}_{z|p}+|\boldsymbol{F}|\alpha\alpha'_{p,z}}{N^{\backslash p,i}_{x=1|p}+K\alpha} \times \frac{N^{\backslash p,i}_{y=0|z}+\gamma^y}{N^{\backslash p,i}_z+2\gamma^y} & \\ \times\frac{N^{\backslash p,i}_{w_{p,i}|z,y=0}+|\boldsymbol{V}|\beta\beta'_{z,w_{p,i}}}{N^{\backslash p,i}_{z,y=0}+|\boldsymbol{V}|\beta} & \text{if } z \in \boldsymbol{F} \\ \frac{N^{\backslash p,i}_{x=1|p}+\gamma}{N_p-1+2\gamma} \times \frac{N^{\backslash p,i}_{z|p}+\alpha}{N^{\backslash p,i}_{x=1|p}+K\alpha} \times 1 \times \frac{N^{\backslash p,i}_{w_{p,i}|z,y=0}+\beta^r}{N^{\backslash p,i}_{z,y=0}+|\boldsymbol{V}|\beta^r} & \text{if } z \in \boldsymbol{E} \end{cases}$$

where Ω is all priors and K is the number of all topics $(|\boldsymbol{S}|+|\boldsymbol{E}|)$. The probability that a feature and a feature value are chosen $(f_{p,i} = z,\ u_{p,i} = j)$ is defined as

(4.10)
$$p(x_{p,i}=1, f_{p,i}=z,\ y=1, u_{p,i}=j$$
$$|w_{p,i}, \boldsymbol{W}_{\backslash p,i}, \boldsymbol{X}_{\backslash p,i}, \boldsymbol{F}_{\backslash p,i}, \boldsymbol{Y}_{\backslash p,i}, \boldsymbol{U}_{\backslash p,i}, \Omega)$$
$$\propto p(x_{p,i}=1|\boldsymbol{X}_{\backslash p,i}, \Omega)p(f_{p,i}=z|\boldsymbol{X}_{\backslash p,i}\boldsymbol{F}_{\backslash p,i}, \boldsymbol{E}_{\backslash p,i}, \Omega)$$
$$p(y_{p,i}=1|z, \boldsymbol{Y}_{\backslash p,i}, \boldsymbol{F}_{\backslash p,i}, \boldsymbol{E}_{\backslash p,i}, \Omega)$$
$$p(u_{p,i}=j|z, \boldsymbol{Y}_{\backslash p,i}, \boldsymbol{F}_{\backslash p,i}, \boldsymbol{U}_{\backslash p,i}, \Omega)p(w_{p,i}|z, j, \boldsymbol{W}_{\backslash p,i}, \boldsymbol{U}_{\backslash p,i}, \Omega)$$
$$\propto \begin{cases} \frac{N^{\backslash p,i}_{x=1|p}+\gamma}{N_p-1+2\gamma} \times \frac{N^{\backslash p,i}_{z|p}+|\boldsymbol{F}|\alpha\alpha'_{p,z}}{N^{\backslash p,i}_{x=1|p}+K\alpha} \times \frac{N^{\backslash p,i}_{y=1|z}+\gamma^y}{N^{\backslash p,i}_z+2\gamma^y} & \\ \times\frac{N^{\backslash p,i}_{j|p}+|\boldsymbol{U}^f|\tau\tau'_{p,z,j}}{N^{\backslash p,i}_{z,y=1|p}+|\boldsymbol{U}^f|\tau} \times \frac{N^{\backslash p,i}_{w_{p,i}|j}+\boldsymbol{V}\rho\rho'_{z,j,w_{p,i}}}{N^{\backslash p,i}_j+\boldsymbol{V}\rho} & \text{if } z \in \boldsymbol{F} \\ 0 & \text{if } z \in \boldsymbol{E} \end{cases}$$

where $|\boldsymbol{U}^f|$ is the number of all possible feature values for the feature f. Regularization is applied to priors for both feature words and feature value words.

Feature Importance The importance of a feature may be useful for a novice customer who wants to know which features are considered important according to reviews. We assume feature importance is determined by how often the features are mentioned in reviews. In SpecLDA, the assignments of feature topics to words can be counted and used as feature popularity. A feature popularity of a feature f is defined as $popularity(f) \propto \frac{N_f}{N_{x=1}}$ where N_f is the number of words assigned with f, and $N_{x=1}$ means the number of words assigned with any features.

5 Experiments

5.1 Data Set

We crawled product reviews and specifications from CNET.com. We chose digital camera category since it is a high-technology product with a lot of features, but our models can be applied to a product category without many features as well. To preprocess the review text, we performed sentence segmentation, word tokenization, and lemmatization using Stanford CoreNLP [15] version 1.3.5. We lowered word tokens and removed punctuations. Then, we removed stopwords provided by Mallet [16], and we pruned word tokens that appear in less than five reviews or more than 30% of the reviews since they are barely informative. We also preprocessed specifications data. We removed feature values that appear in less than five products. Then, we split each feature and feature value text into word tokens by blank, and lowered word tokens.

The resulting data contains 1,153 products that contains specifications and at least one review. The total number of reviews is 11,870, with the total number of sentences being 88,527, where each sentence has 7.74 word tokens on average. The reviews in the same product are concatenated to form a single product document, yielding 1,153 documents with average length being 594.15. In the specifications data, we have a total of 2,320 distinct feature-value pairs S including 124 distinct features $|F|$.

Parameter	DuanLDA	SpecLDA–	SpecLDA
λ	0.3	-	-
α	$\frac{50}{\lvert S\rvert}$	$\frac{50}{\lvert S\rvert+\lvert E\rvert}$	$\frac{50}{\lvert F\rvert+\lvert E\rvert}$
β	0.1	10.0	10.0
β^r	-	0.01	0.01
γ	-	0.5	0.5
δ	-	0.0001	0.0001
ζ	-	0.9	0.9
ϵ_{pp}	-	0.5	0.5
ϵ_{ts}	-	50	50
τ	-	-	0.01
ρ	-	-	10.0
γ^y	-	-	50

Table 1: Hyper-parameter values.

Parameter Setting For all the suggested LDA models, we use five Markov chains after 2,000 Gibbs sampling iterations, where each chain is randomly initialized. Parameter values are empirically set and shown in Table 1, where "-" means not available for the model, and these values are used for all experiments unless specified otherwise.

As shown, SpecLDA– and SpecLDA uses more parameters than DuanLDA. However, many of the additional parameters can be easily set. For example, we can just give a small number to γ, which means very weak supervision on the Bernoulli distribution. The decay factor ζ can be set to a value close to 1.0. We can assign high enough values for initial β and ρ. For DuanLDA, we set β with the highest evaluation scores when the topics hold the original specifications well; we measure topic corruption rate based on KL-Divergence between the original specification words and the estimated topics and try to maintain very similar level of corruption rate for the suggested methods. β^r and δ should be set depending on the sizes of review topics and product-specific topics, respectively, and τ can be set depending on the size of feature-value topics. ϵ_{pp} and ϵ_{ts} can be set depending on how we want to control prior knowledge; if we want the topics more like prior words, we can set ϵ_{pp} close to 1.0 and ϵ_{ts} very high. γ^y is a smoothing parameter on choosing feature or value topics, so greater γ^y results in more even preference.

5.2 Mining Opinions by Sentence Retrieval

Once we learned the topic models, we use the estimated document language models, $p_{lda}(w|p)$, to mine opinion sentences for a specification s. To retrieve text using document language model, we exploit query likelihood model [2], one of the standard ad-hoc retrieval method. Fortunately, in our problem setting, we can take advantage of specifications to filter out some of unrelated sentences; if a sentence t^p is from a product p's reviews, and s is not in p's specifications S_p, we can ignore t^p. The relevance score of t^p for s is thus defined as

(5.11)
$$score(s,t^p) = \begin{cases} 0 & \text{, if } s \notin \boldsymbol{S}_p \\ \prod_{w \in s} \left[(1-\chi)p_{lda}(w|t^p) + \chi p(w|\boldsymbol{W})\right], \text{ o/w} \end{cases}$$

where o/w means "otherwise", $p(w|\boldsymbol{W})$ is a background language model that is estimated by $\frac{count(w)}{|\boldsymbol{W}|}$, where $count(w)$ is the count of w in corpus $\boldsymbol{W}$. χ is a parameter to give a non-zero value to $p_{lda}(w|t^p)$, which is a standard smoothing technique.

Our goal is to mine opinion "sentences", while the suggested topic models estimate "document" language models. We can extend LDA to model each sentence, but it will require too many variables since the number of sentences is usually way greater than the number of documents. Thus, we convert estimations from document-level to sentence-level. Language model $p(w|t^p)$ for a sentence t in a product document p is thus defined as:

(5.12)
$$p_{lda}(w|t^p) = \sum_{z=1}^{K} \frac{\hat{N}_{w|z} + |\boldsymbol{V}|\beta\beta'_{z,w}}{\hat{N}_z + |\boldsymbol{V}|\beta} \frac{\hat{N}_{z|t} + \frac{|t|}{|p|}K\alpha\alpha'_{p,z}}{\hat{N}_t + \frac{|t|}{|p|}K\alpha}$$

where size of topic-document prior $K\alpha$ is re-sized by the proportion of length of t to that of p since topic-document prior depends on the size of document. The same conversion technique is used for other priors that depends on document length in this paper.

Query Expansion by Topic Models In addition to mining opinions based on original specification words, we also try mining opinions with query expansion. One of the advantages of using topic models when topics represent queries is that the estimated topics can be used to expand the original queries. To use a topic as a specification query, we employ KL-divergence model, which is a generalization of the query likelihood model, and its derivation is well explained in [28]. Following the derivation, instead of $c(w, s)$ in formula (5.11), we put $p'(w|s)$, which is a query language model. The query language model we use is the interpolation of the original query model and the estimated topic model, which is defined as

$$p'(w|s) = (1 - \mu)p(w|s) + \mu p(w|\hat{\phi}_s) \tag{5.13}$$

where $p(w|s) = \frac{c(w,s)}{|s|}$, and $p(w|\hat{\phi}_s)$ is the estimated topic language model for the specification s.

5.3 Qualitative Analysis

DuanLDA	SpecLDA–	SpecLDA–_V	SpecLDA
display	digital	lcd	lcd
type	lcd	display	2.5
phtography	2.5	2.5	display
font	display	screen	screen
florescent	huge	type	large
informative	technology	large	inch
channel	expensive	inch	monitor
triple	type	monitor	articulate
resultant	large	phtography	1.8
lcd	outstanding	font	230,000
printout	icon	crack	panel
a10	silver	symbol	bright
colorful	follow	1.8	icon
information	rubber	icon	rotatable
info	phtography	bright	sunlight
horizontal	salesman	brightness	brightness
picture/video	font	articulate	viewfinder
infinite	info	salesman	tilt
2aa	blessing	range	viewer
3.0	symbol	informative	flippable

Table 2: Top 20 words of a topic for a specification (“Display – Type”, “2.5 in. LCD Display”).

Specification Topics The top words in the specification topic (“Display – Type”, “2.5 in. LCD Display”) for each model are listed in Table 2. SpecLDA–_V is a value word prior-added version of SpecLDA– . The word distribution is extracted from $\hat{\boldsymbol{\phi}}$ for DuanLDA, SpecLDA–, and SpecLDA–_V, and $\pi_0\hat{\boldsymbol{\phi}} + \pi_1\hat{\boldsymbol{\omega}}$ for SpecLDA. As shown in the table, words in DuanLDA does not show reasonable relevance to the feature value “2.5 in. LCD Display” since it does not utilize value word priors and the prior size is not regularized. On the other hand, SpecLDA– attracted a few feature value-related words with prior size regularization. SpecLDA–_V, which uses value prior words, drew more value-related words, but it still keeps a few unrelated words. Most of the twenty words SpecLDA attracted seem to be related to the feature value because the feature topic and value topics form a cluster resulting in better topic estimation.

Logitech ClickSmart 310	Canon PowerShot SX10 IS	Canon PowerShot S2 IS
webcam	lens	zoom
cam	zoom	video
video	sx10	s2
image	stamp	movie
digital	20x	cap
computer	video	canon
flash	cap	mode
web	slr	shot
cheap	date/time	sony
figure	superzoom	12x

Table 3: Examples of product-specific topics.

Product-specific Topics In order to inform customers what is special about a product, we capture product-specific topics ψ for each product, and the examples of $\hat{\boldsymbol{\psi}}$ are listed in Table 3. There are several words related to webcams for the product *Logitech ClickSmart 310*, which is actually a webcam that is a rare product category in the data set. *Canon PowerShot SX10 IS* features 20X optical zoom, which is indeed rare (in only five products) and relatively very high performance in the data set, and $\hat{\boldsymbol{\psi}}$ captures related words such as “lens”, “zoom”, “20x”, and “superzoom” quite well. From the top words of *Canon PowerShot S2 IS*, we can see that zoom and video are special. The following sentences from CNET’s editor’s review, which is not included in our data set, support why those words are highlighted for the product.

> The S2’s VGA *movie mode*, which now supports stereo audio, is quite good, with a top resolution of 640x480 at 30fps. Unlike many cameras with similar *movie-capture modes*, the *Canon* lets you use the *zoom*, which operates very quietly, and the IS while capturing *video*.

As shown in the examples, the product-specific topics capture special characteristics of products reasonably well. However, since each product-specific topic is estimated from reviews of a single product, not enough words are assigned with the topic if there are not many review texts for the product.

Most Popular Features
Additional Features – Additional Features
Exposure & White Balance – Shooting Program
Miscellaneous – Included Accessories
Exposure & White Balance – Light Sensitivity
Camera Flash & Flash Modes
Battery – Supported Battery
Memory / Storage – Supported Memory Cards
Lens System – Type
Lens System – Zoom Adjustment
Software

Table 4: Ten most popular features.

Feature Importance By our assumption that more important features are mentioned more frequently in reviews, we compute feature popularity scores and show ten most popular features out of 128 possible features in Table 4. Surprisingly, “Additional Features – Additional Features” is ranked first in the list. The feature has the greatest number of distinct values (170) so that it is present for almost all products, and the word "feature" is one of the most frequent words in the data set. Therefore, many words correlated to the word "feature" are assigned with this feature. Also, the feature "Lens System – Zoom Adjustment" actually contains words more related to zoom capability than zoom adjustment, due to the fact that zoom adjustment is barely mentioned in reviews so that words related to zoom capability are attracted to the topic. This is limitation of our model, and one should consider filtering out those very unpopular features. Other listed features are regarded reasonably important when people purchase a digital camera.

5.4 Quantitative Analysis

5.4.1 Human-labeled Data In order to quantitatively evaluate how well the suggested methods mine opinion sentences for feature values, we need to make a gold standard data set labeled by humans. However, labeling 88,527 sentences by each of multiple annotators is too expensive. A domain expert was asked to choose the twelve most important features by looking at specifications and word counts in the whole corpus. Then, the words in the possible feature-value pairs were used to retrieve candidate sentences (13,671) by the models in Section 4. Each of the candidate sentences was annotated if the sentence is relevant to a specification by three annotators at crowd sourcing service CrowdFlower[3], and the agreement rate was 0.926 and Cohen's kappa coefficient was 0.678, which means quite good agreement among annotators. After pruning non-agreed sentences and queries with less than 20 true relevant sentences, we have 44 queries with 1,251 relevance data on average for each query.

5.4.2 Evaluation Metric We use Mean Average Precision (MAP) as an evaluation metric, which is a mean of average precision for each query. The metric basically measures general retrieval performance for multiple queries, and if we have more relevant documents in high ranks, then the score becomes higher. Specifically, we use MAP@k that computes mean of average precision at the top k retrieved sentences for each query, and we set k as 5, 10, and 20 to see how well the models work in different user satisfaction level.

5.5 Result Analysis We evaluate the sentences retrieved for specification queries. For all the models, we use the original queries, which are concatenated strings of feature words and value words. For SpecLDA– and SpecLDA, we set the number of review topics $|E| = 5$. Table 5 shows evaluation results for the baseline model DuanLDA and our new models SpecLDA– and SpecLDA. † is used to mark models if the improvement is statistically (paired t-test with p=0.05) significant in all measures from the DuanLDA. SpecLDA–, which regularizes β, significantly improves DuanLDA on all measures. SpecLDA outperforms the baseline model significantly on all measures, and it also outperforms SpecLDA–, especially on MAP@20.

	MAP@5	MAP@10	MAP@20
DuanLDA	0.927	0.851	0.780
SpecLDA–†	0.970 (4.6%)	0.894 (5.1%)	0.812 (4.1%)
SpecLDA†	**0.986** (6.4%)	**0.917** (7.8%)	**0.849** (8.8%)

Table 5: MAP evaluation results for finding sentences relevant to a specification. Amount of improvement from DuanLDA is in parenthesis.

Results in Table 6 shows evaluation results when the query language models are expanded by topics. DuanLDA_V is a version of DuanLDA that adds value word priors. Models in the upper part use only feature word priors, and those in the lower part use value word priors as well. Comparing results in Table 5 and those in Table 6, we can see that query expansion by topic models indeed help mining opinion sentences especially with SpecLDA. When value word priors are not used, SpecLDA– significantly outperform DuanLDA. SpecLDA–_V significantly outperform DuanLDA_V, and SpecLDA even improves SpecLDA–_V, which again means SpecLDA's hierarchy structure is effective.

	μ	MAP@5	MAP@10	MAP@20
DuanLDA	0	0.927	0.851	0.780
SpecLDA–†	0.2	**0.978** (6%)	**0.893** (5%)	**0.822** (5%)
DuanLDA_V	0.7	0.964	0.891	0.828
SpecLDA–_V†	1.0	**0.986** (2%)	0.952 (7%)	0.880 (6%)
SpecLDA†	0.7	**0.986** (2%)	**0.969** (9%)	**0.905** (9%)

Table 6: Evaluation results with query expansion. Amount of improvement from DuanLDA (upper part) and DuanLDA_V (lower part) is in parenthesis. † is used to indicate statistical significance on all measures against DuanLDA (upper part) and DuanLDA_V (lower part).

6 Conclusion and Future Work

In this paper, we studied the problem of automatically augmenting product specifications by jointly modeling product reviews and specifications. In specific, we defined the novel problem of relevant sentence retrieval for feature values and suggested a novel approach that is shown to effectively model reviews and specifications. We also demonstrated the inference of feature importance and product-specific words, which may be important for consumers. The potential impact of the augmented specifications is significant since both consumers

[3] http://www.crowdflower.com/

and manufacturers may benefit from them.

While we retrieve all sentences relevant to feature values, one can also suggest retrieving sentences according to sentiment. Retrieving positive, negative, and neutral sentences separately for feature values will help consumers understand the features in a more organized way. Our model also does not consider the time of product reviews being written. A customer feedback on a feature value may be different depending on the time the customer uses it, so disregarding time may result in inconsistency of sentiment on the feature value. If a model considers time, then we can also see how people's opinions change over time and can predict which feature value will be preferred, which may be highly informative for manufacturers. We leave all these interesting problems as future work.

7 Acknowledgments

We are grateful to Huizhong Duan for valuable comments and to Haohong Wang for great support. This work is supported in part by a gift fund from TCL and by the National Science Foundation under Grant Number CNS-1027965.

References

[1] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. *On smoothing and inference for topic models.*, In AUAI, 2009, pp. 27-34.

[2] A. Berger and J. Lafferty. *Information retrieval as statistical translation.* In SIGIR, 1999.

[3] I. Bhattacharya, S. Godbole, and S. Joshi. *Structured entity identification and document categorization: two tasks with one joint model.* In SIGKDD, 2008.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. *Latent dirichlet allocation.* In the Journal of Machine Learning Research, 2003, 3:993-1022.

[5] K. Dave, S. Lawrence, and D. M. Pennock. *Mining the peanut gallery: Opinion extraction and semantic classification of product reviews.* In WWW, 2003.

[6] H. Duan, C. Zhai, J. Cheng, and A. Gattani. *Supporting keyword search in product database: A probabilistic approach.* In Proc. VLDB Endow, Sept. 2013.

[7] T. Hofmann. *Probabilistic latent semantic indexing.* In SIGIR 1999.

[8] M. Hu and B. Liu. *Mining opinion features in customer reviews.* In AAAI 2004.

[9] F. Jelinek. *Interpolated estimation of markov source parameters from sparse data.* In Pattern recognition in practice, 1980.

[10] K. S. Jones and C. J. van Rijsbergen. *Information retrieval test collections* In Journal of documentation. 32(1):59-75, 1976.

[11] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. *An introduction to variational methods for graphical models.* In Machine learning, 37(2):183-233, 1999.

[12] H. D. Kim, K. Ganesan, P. Sondhi, and C. Zhai. *Comprehensive review of opinion summarization.* In Computer Science Research and Tech Reports, 2011.

[13] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications).* In Springer-Verlag New York, Inc., 2006.

[14] Y. Lu and C. Zhai. *Opinion integration through semi-supervised topic modeling.* In WWW. 2008.

[15] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. *The Stanford CoreNLP natural language processing toolkit.* In ACL. 2014.

[16] A. K. McCallum. *Mallet: A machine learning for language toolkit.* http://mallet.cs.umass.edu. 2002.

[17] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. *Topic sentiment mixture: modeling facets and opinions in weblogs.* In WWW. 2007.

[18] B. Pang and L. Lee. *Opinion mining and sentiment analysis.* In Found. Trends Inf. Retr., 2(1-2):1-135, 2008.

[19] I. Peñalver-Martínez, R. Valencia-García, and F. García-Sánchez. *Ontology-guided approach to feature-based opinion mining.* In NLP and Inf. Systems. 2011.

[20] J. M. Ponte and W. B. Croft. *A language modeling approach to information retrieval.* In SIGIR. 1998.

[21] A.-M. Popescu and O. Etzioni. *Extracting product features and opinions from reviews.* In EMNLP. 2005.

[22] T. Tao and C. Zhai. *Regularized estimation of mixture models for robust pseudo-relevance feedback.* In SIGIR. 2006.

[23] I. Titov and R. McDonald. *Modeling online reviews with multi-grain topic models.* In WWW. 2008.

[24] H. Wang, Y. Lu, and C. Zhai. *Latent aspect rating analysis on review text data: a rating regression approach.* In SIGKDD. 2010.

[25] T. Wang, Y. Cai, G. Zhang, Y. Liu, J. Chen, and H. Min. *Product feature summarization by incorporating domain information.* In Database Systems for Advanced Applications. 2013.

[26] X. Wei and W. B. Croft. *Lda-based document models for ad-hoc retrieval* In SIGIR. 2006.

[27] J. Yu, Z.-J. Zha, M. Wang, K. Wang, and T.-S. Chua. *Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews.* In EMNLP. 2011.

[28] C. Zhai. *Statistical language models for information retrieval.* In Synthesis Lectures on Human Language Technologies, 1(1):1-141, 2008.

[29] C. Zhai and J. Lafferty. *A study of smoothing methods for language models applied to ad hoc information retrieval.* In SIGIR. 2001.

[30] C. Zhai, A. Velivelli, and B. Yu. *A cross-collection mixture model for comparative text mining.* In SIGKDD. 2004.

[31] L. Zhou and P. Chaovalit. *Ontology-supported polarity mining.* In Journal of the American Society for Information Science and technology, 59(1):98-110, 2008.