A Sparse Topic Model for Extracting Aspect-Specific Summaries from Online Reviews

Vineeth Rakesh[*]
Arizona State University
Tempe, AZ
vrakesh@asu.edu

Weicong Ding
Amazon
Seattle, WA
weicding@amazon.com

Aman Ahuja
Virginia Tech
Arlington, VA
aahuja@vt.edu

Nikhil Rao
Amazon
San Francisco, CA
nikhilsr@amazon.com

Yifan Sun
Technicolor
Los Altos, CA
yifan.sun@technicolor.com

Chandan K. Reddy
Virginia Tech
Arlington, VA
reddy@cs.vt.edu

ABSTRACT

Online reviews have become an inevitable part of a consumer's decision making process, where the likelihood of purchase not only depends on the product's overall rating, but also on the description of its aspects. Therefore, e-commerce websites such as Amazon and Walmart constantly encourage users to write good quality reviews and categorically summarize different facets of the products. However, despite such attempts, it takes a significant effort to skim through thousands of reviews and look for answers that address the query of consumers. For example, a gamer might be interested in buying a monitor with fast refresh rates and support for Gsync and Freesync technologies, while a photographer might be interested in aspects such as color depth and accuracy. To address these challenges, in this paper, we propose a generative aspect summarization model called APSUM that is capable of providing fine-grained summaries of online reviews. To overcome the inherent problem of aspect sparsity, we impose dual constraints: (a) a spike-and-slab prior over the document-topic distribution and (b) a linguistic supervision over the word-topic distribution. Using a rigorous set of experiments, we show that the proposed model is capable of outperforming the state-of-the-art aspect summarization model over a variety of datasets and deliver intuitive fine-grained summaries that could simplify the purchase decisions of consumers.

CCS CONCEPTS

• **Information systems → Information retrieval**; • **Computing methodologies → Machine learning**;

KEYWORDS

Probabilistic Generative Models; Topic Models; Information Retrieval; Aspect Summarization.

[*]This work was done when the author was an intern at Technicolor Labs, Los Altos, CA, USA.

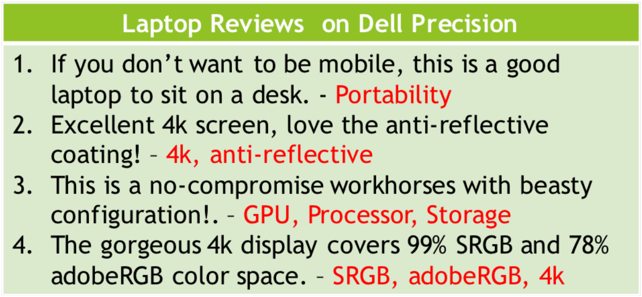
WWW 2018, April 23–27, 2018, Lyon, France
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186069

ACM Reference Format:
Vineeth Rakesh, Weicong Ding, Aman Ahuja, Nikhil Rao, Yifan Sun, and Chandan K. Reddy. 2018. A Sparse Topic Model for Extracting Aspect-Specific Summaries from Online Reviews. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186069

1 INTRODUCTION

Aspect-specific topic detection is an emerging field of research where the goal is to detect fine-grained topics from a large text corpus. For example, consider the set of reviews about the Dell Alienware 15 inch laptop shown in Figure 1. Despite being a popular model, it might not be suitable for every person since the perspective of users vary based on their requirements. For instance, a traveler might be interested in the *portability* aspect of the product, while a gamer might be interested in aspects such as *processor frequency, GPU* and *RAM* and might not give importance to the weight of the laptop. Similarly, a photographer might be interested in high-end displays with great *color accuracy* and *SRGB coverage*, while other consumers might simply look for *budget-friendly* laptop with little importance to such nitty-gritty details. In our example, the Alienware laptop is well acclaimed for its screen and color accuracy; nonetheless, it is not a portable machine. The machine also has great gaming specs, but it is not a budget-friendly laptop. With such varied strengths and weaknesses of the product, it is extremely tedious to manually browse through thousands of user reviews to selectively look for aspects that meet the user's requirements. This emphasizes the need for automated techniques that are capable of mining aspect-specific summaries from user reviews.

A brute force approach to obtain fine-grained aspects is to apply the conventional topic model such as LDA, obtain the topic clusters, and retrieve only those clusters that match the query words. Unfortunately, this technique yields poor results since aspects themselves are sub-topics within an article; hence, they can be extremely sparse. For instance, consider a set of articles about "Global Warming". Let us assume that a person is interested in aspects that talk about the "birth rate of polar bear cubs". Now, since global warming is a very broad topic that covers several other aspects related to the environment, the query of interest (i.e. *polar bear, cubs, birth*) is just a tiny fraction of a vast topic space. Unfortunately, the conventional topic model clusters words from a global perspective, where the query words might get mixed-up with other words from global topics. For



Laptop Reviews on Dell Precision

1. If you don’t want to be mobile, this is a good laptop to sit on a desk. - Portability
2. Excellent 4k screen, love the anti-reflective coating! - 4k, anti-reflective
3. This is a no-compromise workhorses with beasty configuration!. - GPU, Processor, Storage
4. The gorgeous 4k display covers 99% SRGB and 78% adobeRGB color space. - SRGB, adobeRGB, 4k

Figure 1: Example of user reviews about the Dell Alienware 15 inch Laptop.

example, one of the topic clusters for the query *polar bear* could be *smoke, pollution, bear, polar, north*, and *global.* It is quite obvious that such topic clusters do not provide any specific information about the polar bear or their cubs due to other *intruded* words. Another way of modifying the conventional topic model is to first hand-pick sentences that contain the query words and feed this subset of corpus to the model. Unsurprisingly, this method also has some serious shortcomings. First, it leads to severe sparsity of text, which hampers the performance of the LDA model. Second, by throwing away large chunks of text corpus, we lose valuable information about the query itself. Circling back to our example, the aspects *polar bear, cubs*, and *birth* can manifest in different forms such as babies, animals, mammals, creatures etc. Additionally, the description about these aspects need not confine to one single sentence; rather, they could be described over a series of sentences that need not exclusively contain the query words. Therefore, losing such valuable data will lead to incomplete word clusters that could provide very little information about the query. In summary, the classical problems associated with the LDA topic model such as (a) presence of intruded words in coherent word chains, (b) topics with very broad and generic meaning and (c) presence of random noisy words will be greatly amplified if the aforementioned approach is taken to extract fine-grained aspects.

To overcome these challenges, we propose an aspect summarization model called **APSUM** that mines fine-grained aspects for user queries by constricting the document and word topic space to create focused topics. Our goal is to design a model that captures the natural flow of a review writing process. Therefore, we start by asking the question “How does a user write a review?". After observing several reviews from Amazon products and IMDB movie database, we found the following explanation to be a reasonable interpretation of a review writing process. First, a user picks an aspect of interest. Second, he thinks about a sentiment and other aspects relevant to the original aspect of interest. Third, he combines these aspects with other words to create a sentence. For example, consider the *review 2* in Figure 1; here, the user talks about an aspect called *screen* and its *anti-reflective* property (i.e., anti-reflective is a new aspect relevant to the screen) and uses the polarity (or sentiment) *love* to describe this aspect. This interpretation of the review writing process leads us to the following assumptions:

(1) **Assumption 1**: *Every sentence is composed of a narrow range of aspects.* For instance, in *review 2*, we can clearly see that the sentence describes just a couple of aspects (a) the *screen* and (b) the *anti-reflective* property of the screen. Although there can be sentences with multiple aspects, we observed that a majority of them focused on a very narrow range of aspects.

(2) **Assumption 2**: *If we can detect new keywords relevant to the query aspect, these keywords can in turn be used to obtain additional aspects.* To understand this intuition, consider a scenario where a user wants to learn about the *screen* quality of a laptop. Now, if we can somehow detect that the word *4k* is an aspect relevant to the query screen from review 2 (Figure 1), this newly found aspect can then be used to mine other new aspects such as *SRGB* and *Adobe-RGB* from *review 4* since it contains the word *4k*. Consequently, we can cluster the words *screen, SRGB, 4K and Adobe-RGB* as potential aspect words relevant to the query *screen.*

The rest of this paper is organized as follows. We begin by introducing a simple aspect model called M-ASUM in Section 2 and then proceed to explain the proposed APSUM model and the generative process. In Section 3, we explain the collapsed Gibbs sampling and derive the equation for learning the model parameters. The data collection methodology and the results of our experiments are discussed in Section 4. Finally, we review the related works on aspect summarization in Section 5 and conclude our paper in Section 6.

2 MODELING ASPECT SUMMARIES

We begin this section by introducing a simple model that is depicted in Figure 2 (a). Unlike LDA, we do not sample a topic for every word w; instead, a single topic z is drawn for an entire sentence d. The rationale behind this formulation is to mimic our observation that the number of aspects in a sentence is extremely small. After drawing the topic, for every word w, we draw a variable r, which indicates whether a word is an aspect word or a background word. If the word w matches with the query Q or the opinion corpus O we set r to 1 since it is most likely an aspect word. Otherwise, if w is not found in O or Q, then we sample the relevance r from the binomial λ. If $r = 0$, we sample the word from the background distribution ϕ^B; if not, we sample from the word-topic distribution ϕ^{AZ}. In this paper, we consider the sentiment words to be a part of the aspects and do not model them separately. This formulation closely resembles the aspect and sentiment unification model (ASUM) [12] without the sentiment component; hence, we term this model as the modified ASUM (or M-ASUM). Despite being simple, during our experiments, we found that this model was surprisingly good at detecting fine-grained aspects. However, it is not without its flaws. Such brute force approach of constraining the document topic space has severe effects on the smoothness of the word clusters. This is in some sense equivalent to setting the Dirichlet hyper-parameter to zero for achieving sparsity (which is not desirable). Therefore, we propose a sparse aspect summarization model called APSUM that leverages the strengths of M-ASUM, while simultaneously alleviating its weakness.

2.1 Generative Process of APSUM

Figure 2(b) illustrates the plate notation of the APSUM model, which overcomes the shortcomings of M-ASUM using three key components: (a) a document aggregator module, (b) a spike-and-slab

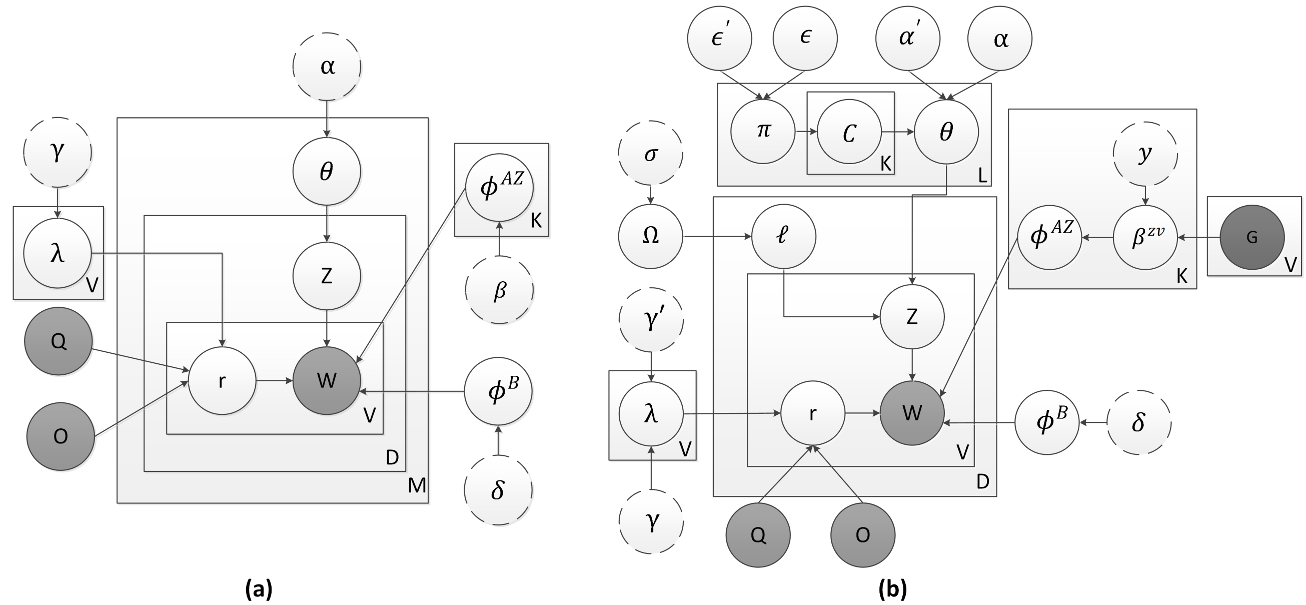


Figure 2: Graphical Structure of (a) the simple aspect model M-ASUM and (b) the proposed aspect summarization model AP-SUM.

prior over the document-topic space and (c) a supervised conditioning over the word-topic hyperparameter. These components are explained in more detail below:

Mitigating the aspect sparsity: In Section 1, we mentioned that the reviews that correspond to a query could be extremely sparse. This essentially translates to the popular problem of the lack of word co-occurrence in short texts. Therefore, we introduce a variable l that acts as a *document aggregator* to overcome this issue. The generative process of the model begins by sampling l for each document. Now, when sampling a topic for a document d, we use the topic distribution of l instead of d.

Constraining the document topic space: The spike and slab technique was originally introduced by Wang et al. [26] to control the navigation of topic mixtures and the word distribution in the probability simplex and later extended to include a weak smoother [13]. In our model (shown in Figure 2(b)), we incorporate this technique using the Bernoulli variable c which introduces the spike by turning-on (i.e., assigning it to 1) or turning-off (i.e., assigning it to 0) a particular topic z. The smoothing is then introduced by hyperparameters α and α'. This ensures that the per-document (i.e., review) topic distribution θ is highly concentrated over a narrow topic space, while simutaneously avoiding document-topic distribution to go ill-defined. So, after sampling l in the previous step, we sample the topic selector c for every topic $z \in K$.

Constraining the word topic space: For obtaining aspects that are focused on the user query, it is important to constrain not only the document-topic proportion, but also the word-topic proportion. To this end, we infuse some supervision into the model in the form of word correlation. In Figure 2(b), this component is shown as the observed variable G. The rationale behind this formulation is simple, if we know that the words *lens* and *zoom* are related, then *we can use this supervised information to relate the topic distribution of these two words.* Therefore, we introduce a *downstream* conditioning on the word-topic smoother β in the form of the variable y. In this way, the word-topic sparsity is naturally infused into the model, while simultaneously avoiding the problem of over-fitting. In [18], the authors use a similar technique to incorporate supervision into the

Table 1: List of notations used in this paper.

Symbol	Description
$D = \{d_i\}$	set of documents, d_i indicates a single document
$V = \{v_j\}$	set of words
r	binary relevance variable, representing r=1 or r=0
C	binary choice variable for *spiking* topic distributions
$L = \{l_i\}$	set of document aggregator
$Z = \{z_i\}$	set of latent topics
K	number of topics
Q, O	observed user query and opinion corpus respectively
θ	document-topic distribution
ϕ^{AZ}	aspect-topic distribution
ϕ^{B}	background word distribution
Ω	document-aggregator distribution
Π	aggregator-topic assignment distribution
λ	word relevance distribution
α, β, δ, σ	hyper-parameters of Dirichlet priors
γ, γ', ϵ, ϵ'	hyper-parameters of Beta priors
n^{ZV}_{k,v^a}	# words v assigned to topic k
$n^{V}_{v^b}$	# background words v^b
$n^{LC}_{l,c}$	# times choice variable c is assigned to an aggreagator l
$n^{LD}_{l,d}$	# aggregator l assigned to a document d
$n^{ZL}_{z,l}$	# words assigned to topic z in aggregator l
$n^{RV}_{r,w}$	# words w assigned to relevance $r = \{0, 1\}$

LDA topic model. The details of this supervision will be explained in the next section.

Continuing with our generative process, after drawing the topic z, for every word w, we draw a variable r, which indicates whether a word is an aspect word or a background word. If the word w matches with the query Q or the opinion corpus O, we set r as 1 since it is most likely to be an aspect word. Otherwise, if w is not found in O or Q, then we sample the relevance r from the multinomial λ. If $r = 0$, we sample the word from the background word distribution ϕ^B, if not we sample from the distribution ϕ^{AZ}. This process is described in Algorithm 1.

3 PARAMETER INFERENCE

With the generative process of the APSUM model, we now derive the collapsed Gibbs sampler for parameter estimation. Recall the likelihood of our model is given by the following equation:

$$\begin{aligned} P(l, z, C, r, w|*) = \int P(l|\Omega)P(\Omega|\sigma)d\Omega \int P(C|\pi)P(\pi|\epsilon, \epsilon')d\pi \\ \int P(r|Q, O, \lambda)P(\lambda|\gamma, \gamma')d\lambda \int P(z|l, \theta)P(\theta|C, \alpha, \alpha')d\theta \\ \int\int P(w|r, z, \phi^{AZ}, \phi^B|\beta)P(\phi^{AZ}|\beta)P(\phi^B|\beta)d\phi^{AZ}d\phi^B \end{aligned} \tag{1}$$

where $*$ refers to the collection of all the hyper-parameters. We estimate the variables C, l, r, and z using the collapsed Gibbs sampling technique as follows.

We begin by sampling the topic selector C. The joint probability distribution of π_l and C_l is given by the following equation:

$$P(\pi_l, C_l|*) \propto \prod_z P(c_{l,z}|\pi_l)P(\pi_l|\epsilon, \epsilon')\frac{I[B_l \in A_l]\Gamma(|A_l|\alpha + K\alpha')}{\Gamma(N_l + |A_l|\alpha + K\alpha')} \tag{2}$$

Here $A_l = \{z : \ c_{l,z} = 1, z = 1, \ldots, K\}$ and $B_l = \{z : \ N_{z,l} > 0, z = 1, \ldots, K\}$. $I(\cdot)$ is the standard indicator function. By integrating out π, the binary variable $c_{l,z}$ is obtained using the following equation:

$$\begin{aligned} P(c_{l,z} = 0|*) \propto (n^{LC}_{l,0} + \epsilon')\frac{I[B_l \in A_l]\Gamma(|A_l|\alpha + K\alpha')}{\Gamma(N_l + |A_l|\alpha + K\alpha')} \\ P(c_{l,z} = 1|*) \propto (n^{LC}_{l,1} + \epsilon)\frac{I[B_l \in A_l]\Gamma(|A_l|\alpha + K\alpha')}{\Gamma(N_l + |A_l|\alpha + K\alpha')} \end{aligned} \tag{3}$$

Second, for each document d, we sample the aggregator l_d. However, from Figure 2(b), we see that l is influenced by the topic distribution θ. To overcome this problem, when sampling l, we assume the topics z as a known variable. This results in the following expression:

$$P(l_d = l|*) \propto \frac{n^{LD}_{l,\neg d} + \sigma}{D - 1 + L\sigma} \frac{\prod_{z \in d}\prod_{j=1}^{N^{ZD}_{z,d}} (n^{ZL}_{z,l,\neg d} + c_{l,z}\alpha + \alpha' + j - 1)}{\prod_{i=1}^{N^{ZD}_{*,d}} (n^{ZL}_{z,l,\neg d} + |A_l|\alpha + K\alpha' + i - 1)} \tag{4}$$

After obtaining l for a document d, we then sample the topic $z_{d,n}$ for each word n according to the following equation:

$$\begin{aligned} P(z_{d,n} = k|Z^{\neg(dn)}, w_{d,n} = v, *) \propto \\ (n^{ZL}_{k,l} + c_{l,k}\alpha + \alpha')\frac{n^{ZV}_{k,v^a,\neg(dn)} + \beta_{k,v^a}}{\sum_{r=1}^{V^a}(n^{ZV}_{k,r^a,\neg(dn)} + \beta_{k,r^a})} \end{aligned} \tag{5}$$

Algorithm 1: Generative process of APSUM model

```
Draw $\phi^B \sim Dirichlet(\delta)$
Draw $\lambda \sim Beta(\gamma, \gamma^{'})$
Draw $\Omega \sim Dirichlet(\sigma)$
Draw $\beta \sim logistic(y; G)$
for each topic $z \in K$ do
    Draw $\phi^{AZ} \sim Dirichlet(\beta_z)$
end
for each aggregator $l$ do
    Draw $\pi_l \sim Beta(\epsilon, \epsilon^{'})$
    for each topic $z \in K$ do
        Draw topic selector $C_{lz} \sim Bernoulli(\pi_l)$
    end
    Draw $\theta_l \sim Dirichlet(\alpha C_l + \bar{\alpha})$
end
for each short document $d \epsilon D$ do
    Sample an aggregator $l \sim Multinomial(\Omega)$
    for each word position $w_i \in d$ do
        Draw $r \sim Bernoulli(\lambda)$
        if $r == 0$ then
            Draw $w_i \sim Multinomial(\phi^B)$
        end
        else
            $w_i \sim Multi(\phi^{AZ})$
        end
    end
end
```

Finally, for each word, the relevance $r_{d,n}$ is sampled as follows:

$$\begin{aligned} P(r_{d,n} = 0|R^{\neg(dn)}, *) \propto (n^{RV}_{0,v,\neg(dn)} + \gamma) \cdot \frac{n^V_{v^b,\neg(dn)} + \delta_{v^b}}{\sum_v^V(n^V_{v^b} + \delta_v)} \\ P(r_{d,n} = 1|R^{\neg(dn)}, *) \propto (n^{RV}_{1,v,\neg(dn)} + \gamma') \cdot \frac{n^{ZV}_{k,v^a,\neg(dn)} + \beta_{k,v^a}}{\sum_v^V n^{ZV}_{k,v^a,\neg(dn)} + \beta_{k,v^a}} \end{aligned} \tag{6}$$

The above expression marks the end of our Gibbs sampling process and we proceed with the methodology of achieving the word-topic sparsity. It should be noted that the negative log-likelihood $p(w|z, \beta)$ of APSUM remains similar to the LDA topic model and is defined as follows:

$$\begin{aligned} L_\beta = \sum_{z=1}^{K}[log\Gamma(\beta_z + n^k_z) - log\Gamma(\beta_z)] \\ + \sum_{z=1}^{K}\sum_{v}^{V}[log\Gamma(\beta_{zv}) - log\Gamma(\beta_{zv} + n^{kv}_{zv}) \end{aligned} \tag{7}$$

Now, instead of using the symmetric prior β, we modify it using a topic dependent coefficient y (shown in Figure 2(b)) as follows:

$$log\, p(\beta) = \frac{-1}{2\lambda^2}\left[\sum_{v,v',z} G_{v,v'}(y_{zv} - y_{zv'})^2\right] \tag{8}$$

where $G_{v,v'}$ is the observed relationship between the words (i.e., *lens* and *zoom*). Using equations (7) and (8), the objective function

Table 2: Characteristics of the Review Dataset

Reviews	#Docs	Source	Queries
Restr	254	SEMVAL	*wine, sushi, service, pizza*
Hobbit	1k	IMDB	*jackson, smaug, legolas, dwarf*
CivilWar	1.3k	IMDB	*panther, spider-man, plot, fight*
Camera	5k	Amazon	*picture, lens, battery, autofocus*
HomTh	5k	Amazon	*wireless, woofer, pandora,vizio*

is defined as follows:

$$\underset{y_{zv}}{\operatorname{argmin}}\,[L_\beta - logp(\beta)] \tag{9}$$

From the above function, one can realize that by optimizing over y, we *dynamically change the prior β with the aid of the observed (or supervised) variable G.*

Construction of the linguistic graph *G*: The observed variable G in Figure 2, is created using two kinds of supervision: (a) constructing a linguistic dependency graph using the Stanford NLP module [7] and (b) a simple entity relationship.

The dependency graph from the natural language processing domain provides the grammatical relationships between words to induce extraction rules. For example in the sentence "Nikon D500 has a great lens", we can extract *lens* and *D500* as potential aspect words utilizing the *aspect-aspect* relationship (i.e., AA-Rel) induced by the following dependency structure: $lens \rightarrow$**obj** $\rightarrow has \leftarrow$ **subj** $\leftarrow D500$. There are many such rules for aspect extraction, but in this paper, we restrict our explanation to this simple example since creating these rules is not the main focus of our work. Instead, we simply utilize existing studies [11, 31] and python NLP tools[1] to extract such potential aspect words. Entity extraction on the other hand is a simple process and we use the same python module to *extract just the entities from the review sentences* and treat them as potential aspect words.

4 EXPERIMENTS

In this section, we perform a rigorous series of quantitative and qualitative experiments over various datasets and test cases to evaluate the proposed model. Usually the document-topic and word-topic hyperparameters in topic models are set to 0.1 and 0.01 respectively; therefore, we follow the same convention. The model-specific hyperparameters η, σ, λ are set to 0.1, $\beta = 0.01$, $\delta = 0.001$, and the weak smoother $\lambda' = 0.00001$. These values are decided based on trial and error method after performing some initial experiments and manually judging the quality of the aspects produced by the APSUM model. The number of topics are varied between $100 - 150$ and the aggregator variable l is also varied between $150 - 250$ depending on sparsity of the query. The iteration count is set to 250 and the optimization over the word smoothing coefficient y is performed after a burn-in period of 50 iterations. The implementation of M-ASUM and APSUM models can be downloaded from our Github repository[2].

4.1 Datasets Used

For our experiments, we obtained datasets from three different domains. The information about these datasets are detailed as follows:

(a) **Restaurant reviews from SEMVAL:** Semantic Evaluation (SEMVAL) [24] is a popular workshop on evaluations of computational semantic analysis systems, which provides annotated datasets for various information retrieval problems. For our experiments, we use the restaurants review data from the *Sentiment track, Task 12* of SEMVAL 2015 [22].

(b) **Movie reviews from IMDB:** We used IMDB's python API[3] to crawl movie reviews from the internet movie database. For this paper, we select the following movie reviews: (a) *Hobbit: The Desolation of Smaug*, and (b) *Captain America: Civil War.*

(c) **Product reviews from Amazon:** The 50 domain online review dataset from Amazon is another text corpus that is popular with the information retrieval community [5]. We evaluate our model on two product categories, namely *Camera* and *Home Theatre.*

For each dataset, we select four different queries to measure the performance of the APSUM model. The basic statistics and the queries for each dataset are summarized in Table 2. We adopt the conventional pre-processing steps which includes tokenization, removing stop words, lemmatization and removing vocabularies with word count fewer than five words.

4.2 Comparison Methods

We compare the performance of the proposed model with the following baseline methods:

(1) **LDA:** Our first candidate for comparison is the classic Latent Dirichlet Allocation (LDA) model [2]. For every dataset, we run the LDA model by setting the hyperparameters α as 0.1 and β as 0.01 and the number of topics as 70. The resulting topic clusters are then manually evaluated to see whether the word clusters are relevant to the target aspect (or query).

(2) **MG-LDA:** Proposed by Titov et. al., the multi-grain model [25] is one of the popular works on detecting aspect-specific topics from online reviews where aspect granularity is achieved by modeling both global and location topic distribution. The MG-LDA uses four hyperparameters $\gamma, \alpha^{gl}, \alpha^{loc}$ and α^{loc}. In our experiments, all these parameters are set to 0.1 and number of topics as 100.

(3) **M-ASUM:**. As mentioned in Section 2, the simple aspect model (M-ASUM) proposed in this paper (Figure 2) is a variation of the *aspect sentiment unification model* (ASUM) [12], where the topics are sampled for an entire sentence instead of the conventional per-word sampling. In our experiments, parameters γ, δ are set to 0.1, β as 0.001 and the number of topics as 100.

(4) **Targeted-Topic Model (TTM):** The TTM is the state-of-the-art model for aspect-based topic summarization [27]. Therefore, in this paper, we choose the TTM model as the prime candidate for comparison. The model has seven hyper-parameters which are set as follows: $\gamma = \alpha = 1, p = q = 1, \beta^{ir} = \delta = 0.001$ $,\epsilon = 1.0 \times 10^{-7}$ and number of topics as 10.

The parameters of the above baselines were chosen based on trial and error. However, we noticed that for most of the scenarios the

[1] https://github.com/dasmith/stanford-corenlp-python
[2] https://github.com/VRM1/WWW18
[3] https://pypi.python.org/pypi/imdbpie/

Table 3: Performance comparison of ASUM in terms of the precision scores.

Dataset	**Aspects**	**LDA**			**MG-LDA**			**M-ASUM**			**TTM**			**APSUM**		
		p@5	p@10	p@20	p@5	p@10	p@20	p@5	p@10	p@20	p@5	p@10	p@20	p@5	p@10	p@20
Restr	wine	0.36	0.3	0.13	0.57	0.43	0.21	0.58	0.56	0.41	0.66	0.61	0.38	0.76	0.72	0.41
	sushi	0.34	0.31	0.15	0.61	0.53	0.36	0.66	0.62	0.35	0.57	0.52	0.29	0.63	0.57	0.3
	service	0.29	0.22	0.1	0.59	0.43	0.22	0.52	0.49	0.21	0.62	0.53	0.33	0.71	0.63	0.31
Hobbit	jackson	0.46	0.35	0.19	0.49	0.48	0.19	0.41	0.42	0.22	0.66	0.56	0.41	0.69	0.61	0.46
	smaug	0.51	0.45	0.22	0.71	0.63	0.41	0.73	0.69	0.31	0.76	0.71	0.42	0.88	0.83	0.49
	sauron	0.41	0.36	0.2	0.71	0.59	0.39	0.66	0.61	0.35	0.71	0.65	0.4	0.79	0.77	0.43
Civil War	panther	0.56	0.44	0.24	0.69	0.51	0.22	0.69	0.55	0.38	0.81	0.77	0.45	0.85	0.79	0.46
	Spider-man	0.48	0.36	0.19	0.73	0.67	0.34	0.71	0.68	0.39	0.79	0.71	0.44	0.82	0.79	0.4
	plot	0.41	0.37	0.18	0.66	0.52	0.26	0.68	0.61	0.31	0.69	0.65	0.38	0.73	0.67	0.43
Camera	picture	0.53	0.46	0.21	0.65	0.58	0.25	0.61	0.55	0.41	0.77	0.7	0.41	0.77	0.73	0.44
	autofocus	0.25	0.18	0.11	0.58	0.48	0.28	0.6	0.61	0.35	0.65	0.59	0.35	0.7	0.71	0.38
	lens	0.22	0.23	0.09	0.6	0.51	0.25	0.63	0.61	0.31	0.61	0.62	0.38	0.71	0.66	0.38
HomTh	wireless	0.19	0.18	0.1	0.58	0.46	0.22	0.51	0.45	0.3	0.58	0.54	0.32	0.66	0.55	0.4
	woofer	0.26	0.21	0.12	0.66	0.53	0.31	0.67	0.65	0.32	0.65	0.61	0.4	0.73	0.64	0.41
	pandora	0.18	0.15	0.08	0.51	0.47	0.29	0.49	0.42	0.28	0.49	0.35	0.25	0.54	0.41	0.28
average score		0.36	0.3	0.15	0.62	0.52	0.28	0.61	0.57	0.32	0.67	0.6	0.37	0.73	0.67	0.4
APSUM performance gain		**0.37**	**0.37**	**0.25**	**0.11**	**0.15**	**0.12**	**0.12**	**0.1**	**0.08**	**0.06**	**0.07**	**0.03**			

models gave the best results with the default value that was set by the authors.

4.3 Evaluation Methodology

Judging the Topic Quality: Topic models are typically evaluated using popular methods such as perplexity or the likelihood of held-out data; nonetheless, researchers have shown that these automated methods of evaluation does not translate to the actual human interpretability of topics [4]. Therefore, in our paper, we adopt the following techniques to judge the quality of topics produced by APSUM: (a) human judgment and (b) topic coherence. In order to perform the human judgment, for each domain, (i.e movie reviews, product and restaurant reviews) we selected three students who are experts in judging topics related to movies and three of our collaborators who are experts in judging product-related topics. The quality of the topics were decided based on the majority voting scheme.

4.3.1 Evaluation Metrics. For our first evaluation, we use a normalized version of the precision metric that was proposed by Wang et al. [27]. The precision score for a model m is defined as follows:

$$P_m@T = \frac{\sum_{z=k}^{K_m} \#Rel(Q_z)}{\sum_{z=k}^{K_u} \#MaxRel(Q_z)} \tag{10}$$

where K_m is the set of aspects (or topics) that matches the user's query of interest, $\#Rel(Q_z)$ is the number of words that are relevant to the aspect z, K_u is total set of unique aspects from all models that are relevant to the user's query, and $\#MaxRel(Q_z)$ is the maximum number of words that are relevant to the user query. It should be noted that this count is obtained from the model that provides the best aspect for the query.

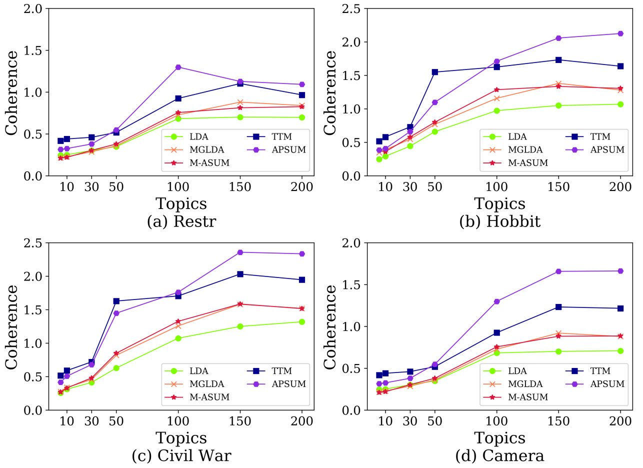
The second evaluation measure is the topic coherence, which is defined as follows:

$$coherence(V) = \sum_{v_i, v_j} score(v_i, v_j, \epsilon) \tag{11}$$

where V is the vocabulary and ϵ is the smoothing factor. The $score(v_i, v_j, \epsilon)$ signifies the mutual information between two words and can take many different forms. The most poplar ones are the UMass measure [15] and the UCI measure [17]; in this paper, we choose the latter.

4.3.2 Quantitative Evaluation. Table 3 summarizes the results of our experiments, which reveal several interesting outcomes. First, all four models show a clear improvement over the standard LDA; thereby, proving that the conventional topic model is not suitable for detecting fine-grained aspects due to its tendency to generate global topics. Second, it is also quite obvious that the APSUM outperforms every other model by producing better precision scores. When comparing with TTM, APSUM has a gain of 6-7% over top-5 and top-10 words, but this gain slides down to just 3% when considering top-20 words. The reason for such diminishing gains can be attributed to the composition of the data, where most reviews (especially, the product data) are extremely short, and cover a narrow range of aspects with very limited vocabulary. For example, in the restaurant dataset, only six or seven reviews mentioned something about *Sushi* and more importantly the description was limited to just 4-5 lines and over 70% of them had strong overlap of words related to aspects such as *starter, appetizer, tuna, asian and service.*

Comparing the proposed model to MG-LDA and M-ASUM, we see a performance increase of upto 15% and unsurprisingly, the largest gain achieved by APSUM is over the standard LDA model with about 37% improvement over the top 10 ranked words. We also tried to increase the number of topics for both MG-LDA and M-ASUM from 50 to 100 to see whether there is an increase in the aspect quality. Although this definitely resulted in mining more aspects, the resultant topic space was too noisy and the human



(a) Restr (b) Hobbit (c) Civil War (d) Camera

Figure 3: Comparison of Topic Coherence.

judgment became too tedious. Table 3 also shows another interesting trend where MG-LDA seems to perform better than M-ASUM for the top-5 words; however, for top-10 and top-20 words, this outcome is reversed, where the former outperforms the latter. As mentioned in the previous section, LDA had the tendency to constantly produce a large set of globally related topics that were incoherent with the aspect of interest. Due to space constraints we only show the results of three queries, but the average precision scores was calculated using the outcome of the fourth query.

Analyzing the topic coherence: The coherence scores of APSUM shown in Figure 3 reveals some intriguing similarities between the precision scores obtained using human judgment. First, the overall performance of APSUM is significantly better than other models across all datasets and the best coherence score is achieved over the movie dataset due to its rich word co-occurrence information. However, one can also observe that TTM supersedes our model for lower topic counts (i.e., between 10-50). This is mainly due to the structure of the graphical model proposed by the authors of TTM. That being said, our model significantly beats TTM when the number of topics exceeds hundred. Second, MG-LDA and M-ASUM have very similar coherence scores and LDA trails behind all other models; thus establishing a surprising analogy with the precision scores shown in Table 3. Due to space constraints, we exclude the results of the home theater reviews, but in our testing, the performance was very similar to the dataset on camera reviews.

Effect of linguistic supervision: We conclude this section by illustrating the effect of supervision in Figure 4, where x-axis denotes the different supervised information, and y-axis is the precision at K ($p@K$), which is calculated using the same human judgment. The term *s-prior* signifies the unsupervised version of our model that uses symmetric priors for all hyperparameters and the terms *D-graph* and *Ent* denote the supervision using dependency graphs and entities, respectively (refer to Section 3). The figure clearly shows that the linguistic supervision in-terms of the dependency graph provides a reasonable boost to the performance of APSUM to mine aspects that are better correlated. On the other hand, the entity-type supervision is not as good as the D-graph supervision due to its simplistic nature. It looks like simply detecting entities in sentences does not convey sufficient information about the aspects themselves. Obviously, apart from the supervised information, the topic count plays a significant role in determining the precision. Except for the restaurant review dataset, a topic count of 150 yields the best performance. This is because, in restaurant dataset, the number of documents (i.e. reviews) about a specific item is extremely sparse; therefore, increasing the number of topics simply introduces noise, thereby rendering the supervision moot.

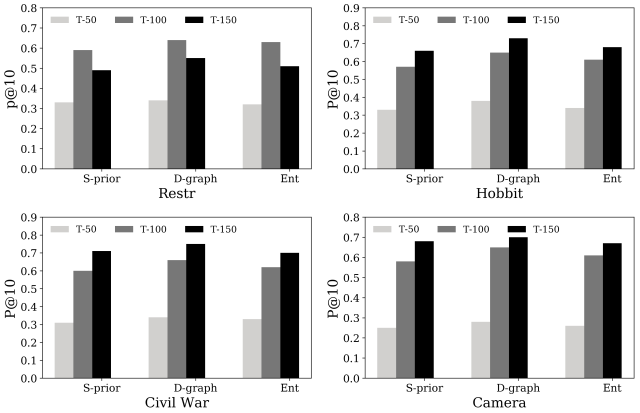


Figure 4: Effect of linguistic supervision.

4.3.3 Qualitative Evaluation. In this section, we perform qualitative analysis of the proposed model by showing the actual aspect summaries and analyze their quality from a perspective of human understanding. Due to space constraints, it is not feasible to show the outcome of every model and query. Consequently, besides APSUM, we choose two other models that produced the best results in our quantitative evaluation, namely TTM and M-ASUM. Table 4 shows the aspects produced by these models over the queries, *battery* and *wireless*, where the words marked in red denote the *intruded* (or noisy) words. From the results, it is quite apparent that the aspects produced by APSUM are very focused on the target query and more importantly, the word clusters under each aspect are extremely coherent in conveying a unified theme. For instance, the aspect *capacity* is highly relevant to the query *Battery* and the words *life, charge, average, screen*, etc., signify certain attributes of this aspect. Similarly, most words associated with the queries *Speaker* and *Setup* are highly accurate in describing the characteristics of these aspects. The same cannot be said about TTM, where, for some cases the model performs extremely well, while for others it is too noisy and fails to convey a coherent theme. For example, the words *card, video, dslr* and *memory* appears to convey some meaning about the aspect on *memory cards*; however, the intruded words *canon, software, decide* etc., have very little correlation with this aspect. This makes this topic cluster extremely generic and difficult to interpret. Finally, the example also shows that the topics produced by M-ASUM are reasonably good. However, when compared to APSUM and TTM, it definitely tends to have more intruded (or irrelevant) words. In fact, in some cases such as the query about *Wireless*, there are many random words like *money, samsung, plasma*, etc.

We now analyze the aspects from the movie dataset. A key trait that separates the movie from the product dataset is the homogeneity of reviews. In other words, the reviews about cameras are

Table 4: Qualitative comparisons of the aspects produced by APSUM for the product review dataset from Amazon.

Domain: Camera, Query:Battery						Domain: Home Theater, Query:Wireless					
APSUM			TTM		M-ASUM	APSUM		TTM		M-ASUM	
Capacity	**Type**	**Video**	**Generic**	**Size**	**Capacity**	**Speaker**	**Setup**	**Speaker**	**Setup**	**Router**	**Generic**
life	hour	extra	canon	picture	picture	wireless	wall	wireless	subwoofer	wireless	money
capacity	aa	video	card	video	big	subwoofer	setup	speaker	unit	subwoofer	samsung
charge	pack	action	video	feel	aa	speaker	unit	rear	add	theater	wireless
average	lithium	shot	dslr	hand	action	powered	mount	good	angle	router	plasma
screen	rechargable	personal	memory	nikon	power	sound	wireless	surround	arrangement	receiver	star
spare	compartment	summer	software	big	huge	great	bracket	added	direction	capability	inch
removable	charger	roll	decide	hold	kodak	corner	cord	amazing	deal	ghz	hdmi
minute	extra	shutter	upgrade	easy	folk	bass	router	apartment	beautiful	internet	bass
advantage	style	dvd	return	film	medium	watt	little	authorized	folk	comcast	feature

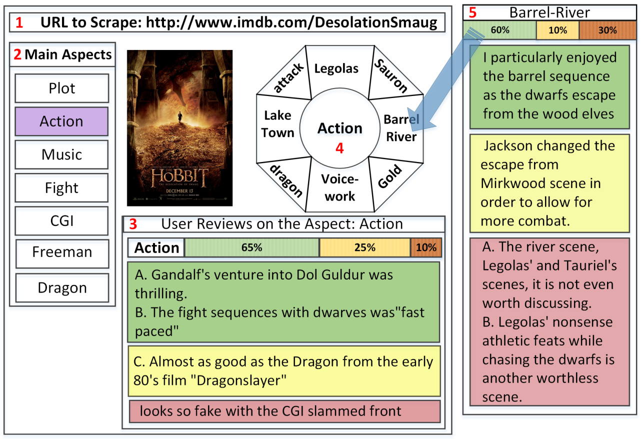
Table 5: Qualitative comparisons of the aspects produced by APSUM for the movie review dataset from IMDB.

Domain: Desolation of Smaug, Query:legolas						Domain: Captain America Civil War, Query: Fight					
APSUM			TTM		M-ASUM	APSUM		TTM		M-ASUM	
Love	**Combat**	**Chase**	**Love**	**Generic**	**Love**	**Airport**	**Team**	**Airport**	**Action**	**Airport**	**Antman**
tauriel	legolas	dwarf	legolas	bloom	legolas	scene	america	fight	good	fight	stuff
legolas	orcs	barrel	elf	orlando	tauriel	action	captain	character	iron	scene	antman
dwarf	sequence	orcs	love	elf	horse	airport	cap	film	movie	airport	masterpiece
kili	scene	river	kili	film	orcs	sequence	team	airport	cap	epic	funny
elf	orc	chase	scene	book	female	fight	bucky	scene	bucky	end	nerd
triangle	head	legolas	triangle	story	thranduil	choreography	tony	final	soldier	bucky	huge
love	great	ride	dragon	thranduil	love	great	final	battle	new	decade	beautiful
line	combat	water	return	action	need	seat	ironman	emotional	spiderman	menace	zack
relationship	horse	able	good	lotr	relationship	edge	reason	epic	brother	funny	natasha

not about a specific product; instead, it is a combination of varied product types that include compact cameras, DSLRs, and binoculars from different brands such as *Canon, Nikon,* and *Sony*. Consequently, the aspect summaries of this dataset are not necessarily focused on any particular merchandise. The movie dataset on the other hand, consists of reviews that talk about a *specific movie*; therefore, the aspect summaries are focused towards the characteristics of a movie. Table 5 demonstrates this outcome using the queries *Legolas* over the movie “Hobbit: The Desolation of Smaug” and the attribute *fight* over the movie “Captain America Civil War”.

The film potrays the character *Legolas* in three main scenes: (a) a scene that depicts Legolas to have possessive feelings over the relationship between the elf Tauriel, and the dwarf Kili. (b) a well acclaimed chase between the orcs, the elves and the dwarfs on a fast flowing river and (c) a fight sequence where Legolas chases an orc named Bolg. All three scenes are summarized under the aspects *Love*, *Combat* and *Chase*, respectively. The TTM model also produces some good aspect summaries, but the word features that describe these aspects are noisier when compared to APSUM. For instance, the aspect *love* has some intruded words such as *dragon* and *return*, which do not coincide with the main theme. Additionally, we also observe a very *generic* topic (i.e., global topic), which basically has all the popular words from the movie and such topics convey little to no meaning. Similar arguments can be made over the results of the query *fight*. The movie depicts an intense face-off between Iron-man and Captain America at the *airport*, which was acclaimed by many critiques. Table 5, shows that both APSUM and TTM are very good in summarizing this aspect. Apart from this scene, APSUM also produces an aspect called *team*, which summarizes the main characters involved in the climax fight scene. Alternatively, TTM is able to reveal some sentiment words related to the *action* aspect, while M-ASUM simply produces a noisy topic cluster that seems to be related to the character *ant-man*. Readers should note that we tried to retrieve the same aspects from all models for fair comparison. However, this was not feasible since the topics produced by models greatly vary. For instance, in this example, TTM never produced a topic that was relevant to the *type* of *battery* (i.e., aa, lithium, etc.) while APSUM did not summarize any aspect related to the size. In summary, from these qualitative examples, it is clear that APSUM outperforms other baselines and the state-of-the-art aspect model. Although we are unable to show the results of all the queries, in our rigorous testing, we found that APSUM produced focused and human interpretable aspects even on sparse datasets due to three key components: (1) the document aggregator l that mitigates the problem of word co-occurrence, (2) the spike-and-slab prior constraining the document-topic space and (3) the downstream conditioning y_{kv} on the topic smoother β that was discussed in Section 2.

4.3.4 Visual Interface of Aspect Summaries. Although we performed rigorous evaluation of our model using various test cases, one might still ask the question, “how can this model be useful to the end-user?”. To answer this question, we provide a visual



A. Gandalf's venture into Dol Guldur was thrilling.
B. The fight sequences with dwarves was"fast paced"

C. Almost as good as the Dragon from the early 80's film "Dragonslayer"

looks so fake with the CGI slammed front

I particularly enjoyed the barrel sequence as the dwarfs escape from the wood elves

Jackson changed the escape from Mirkwood scene in order to allow for more combat.

A. The river scene, Legolas' and Tauriel's scenes, it is not even worth discussing.
B. Legolas' nonsense athletic feats while chasing the dwarfs is another worthless scene.

Figure 5: A visual interface of aspect-specific topic summarization system.

interface of our system in Figure 5. Different blocks of this interface are numbered in red color. In block 1, the user enters a URL of an item to scrape its reviews. In our example, the item is the movie "Hobbit: The Desolation of Smaug". The interface then presents the user with different global aspects (block 2) obtained from the word-topic proportions ϕ^{AZ} of the APSUM model (Figure 2). Now, let us assume the user clicks on the main aspect named "Action". The interface then provides the sentiments and the original reviews about *action* in block 3. In addition to this, users can also view and click the sub-aspects related to the main aspect in block 4. For instance, if the user clicks on the aspect *barrel-river*, the interface then displays the sentiment proportions and the user reviews that are specific to this sub-aspect (i.e., block 5). Readers should note that although our approach does not exclusively model sentiments, we can still obtain the aspect-specific sentiment distribution by simply detecting the polarities of top words in the word-topic proportions. The sentences for a given aspect can be obtained by mapping the document-topic proportion θ to ϕ^{AZ}.

5 RELATED WORKS

Aspect-specific topic summarization of textual reviews is an emerging field of research. Therefore, there are very few research works that exclusively tackle this problem [8, 12, 27–29, 33]. That being said, the techniques used in formulating such models are closely related to sparse topic models that operate on microblogging data from Twittter, Tumblr, Friendfeed, etc. One of the earliest works on granularizing LDA to detect fine-grained aspects can be seen in [25]. In their work, the authors propose a multi-grain topic model called MG-LDA that extends the standard LDA model to generate global topics at a document-level and local topics on a sentence-level. Later works on aspect detection incorporate sentiment words as a part of their joint modeling framework [3, 12, 21, 32]. In [16], the authors provide a comprehensive summary of various aspect and opinion summarization models. In a recent work, Yang et al. [30] leverage metadata about reviews such as gender, location and age to propose a user-aware topic model that jointly models aspect and meta-data about the users and topics. In [10], the authors introduce a supervised topic model that utilizes the overall rating of reviews to treat the documents as a bag-of-opinion pair; where, each pair consists of an aspect and an opinion associated with that aspect.

Apart from topic models, another popular way of detecting aspects is to use linguistic techniques from the domain of natural language processing (NLP). The NLP techniques can be as simple as extracting aspects based on frequently occurring noun phrases (NP) [1, 19] to more comprehensive techniques that involve building dependency grammar structures. For instance, the authors of [31] and [20] propose an aspect and entity extraction module that uses several grammar rules [7] to create dependency graphs between words. One of the popular works by Hu et al. [11] provide feature-based summaries of customer reviews on products such as digital camera, cellular phones and Mp3 players. The popular workshop on semantic evaluation (SEMVAL) provides an exclusive track on aspect-based sentiment detection, where several researchers compile heuristic techniques to mine aspect and sentimens [6, 9]. More recently, in [14], the authors leverage the prior knowledge from several other product domains (e.g., reviews of products from electronic category) to extract aspects of the target product. A comprehensive summary of aspect-level topic and sentiment detection can be seen in [23].

Despite such recent works on aspect-specific topic summarization, there is still room for several improvements since detecting fine-grained topics from large textual corpus is still an open problem. The research that is closest to our work is the targeted topic model (TTM) [27] where the authors use the spike-and-slab prior over the word-topic space. However, for achieving topic sparsity, it is important to perform both upstream (i.e., the document-topic simplex) and downstream conditioning (i.e., the word-topic simplex). Additionally, our method allows to incorporate supervision in the form of human annotation, linguistic dependency graphs and other information from external document corpus to improve the quality of summarized aspects. The results of our model clearly reveal the effectiveness of this approach by producing superior performance over TTM.

6 CONCLUSION

In this paper, we proposed a generative topic model, called APSUM, that is capable of retrieving and summarizing fine-grained aspects from online reviews. To achieve aspect sparsity in the word distribution, we performed a joint modeling of three different components: (1) a sentence aggregator to overcome the sparsity of word co-occurrence, (2) a spike-and-slab prior to introduce sparsity in document-topic space, while avoiding over-fitting using a smoother and (3) a supervised conditioning over the hyperparameter β to infuse word-topic sparsity. Using extensive set of experiments, and a variety of datasets from different domains, we showed that our model outperformed all the baselines and the state-of-the-art aspect summarization model in both quantitative and qualitative evaluations.

Acknowledgements

This work was supported in part by the National Science Foundation grants IIS-1619028, IIS-1707498 and IIS-1646881.

REFERENCES

[1] Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A Reis, and Jeff Reynar. 2008. Building a sentiment summarizer for local service reviews. In *WWW workshop on NLP in the information explosion era*, Vol. 14. 339–348.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[3] Samuel Brody and Noemie Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 804–812.

[4] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*. 288–296.

[5] Zhiyuan Chen and Bing Liu. 2014. Topic Modeling using Topics from Many Domains, Lifelong Learning and Big Data. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 703–711.

[6] Orphée De Clercq, Marjan Van de Kauter, Els Lefever, and Véronique Hoste. 2015. Applying hybrid terminology extraction to aspect-based sentiment analysis. In *International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, 719–724.

[7] Marie-Catherine De Marneffe and Christopher D Manning. 2008. *Stanford typed dependencies manual*. Technical Report. Technical report, Stanford University.

[8] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 193–202.

[9] Aitor Garcıa-Pablos, Montse Cuadros, and German Rigau. 2015. V3: unsupervised aspect based sentiment analysis for SemEval-2015 Task 12. *SemEval-2015* (2015), 714.

[10] Zhen Hai, Gao Cong, Kuiyu Chang, Peng Cheng, and Chunyan Miao. 2017. Analyzing Sentiments in One Go: A Supervised Joint Topic Modeling Approach. *IEEE Transactions on Knowledge and Data Engineering* 29, 6 (2017), 1172–1185.

[11] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 168–177.

[12] Yohan Jo and Alice H Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 815–824.

[13] Tianyi Lin, Wentao Tian, Qiaozhu Mei, and Hong Cheng. 2014. The dual-sparse topic model: mining focused topics and focused terms in short text. In *Proceedings of the 23rd international conference on World wide web*. ACM, 539–550.

[14] Qian Liu, Bing Liu, Yuanlin Zhang, Doo Soon Kim, and Zhiqiang Gao. 2016. Improving Opinion Aspect Extraction Using Semantic Similarity and Aspect Associations.. In *AAAI*. 2986–2992.

[15] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 262–272.

[16] Samaneh Moghaddam and Martin Ester. 2012. On the design of LDA models for aspect-based opinion mining. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 803–812.

[17] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. 2010. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*. ACM, 215–224.

[18] James Petterson, Wray Buntine, Shravan M Narayanamurthy, Tibério S Caetano, and Alex J Smola. 2010. Word features for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*. 1921–1929.

[19] Ana-Maria Popescu and Orena Etzioni. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*. Springer, 9–28.

[20] Soujanya Poria, Erik Cambria, Lun-Wei Ku, Chen Gui, and Alexander Gelbukh. 2014. A rule-based approach to aspect extraction from product reviews. In *Proceedings of the second workshop on natural language processing for social media (SocialNLP)*. 28–37.

[21] Md Mustafizur Rahman and Hongning Wang. 2016. Hidden topic sentiment model. In *Proceedings of the 25th International Conference on World Wide Web*. 155–165.

[22] José Saias. 2015. Sentiue: Target and aspect based sentiment analysis in semeval-2015 task 12. Association for Computational Linguistics.

[23] Kim Schouten and Flavius Frasincar. 2016. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering* 28, 3 (2016), 813–830.

[24] SEMVAL. [n. d.]. International Workshop on Semantic Evaluation. http://alt.qcri.org/semeval2016/

[25] Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 111–120.

[26] Chong Wang and David M Blei. 2009. Decoupling sparsity and smoothness in the discrete hierarchical dirichlet process. In *Advances in neural information processing systems*. 1982–1989.

[27] Shuai Wang, Zhiyuan Chen, Geli Fei, Bing Liu, and Sherry Emery. 2016. Targeted Topic Modeling for Focused Analysis. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

[28] Shuai Wang, Zhiyuan Chen, and Bing Liu. 2016. Mining Aspect-Specific Opinion using a Holistic Lifelong Topic Model. In *Proceedings of the 25th International Conference on World Wide Web*. 167–176.

[29] Yao Wu and Martin Ester. 2015. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 199–208.

[30] Zaihan Yang, Alexander Kotov, Aravind Mohan, and Shiyong Lu. 2015. Parametric and non-parametric user-aware sentiment topic models. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 413–422.

[31] Lei Zhang and Bing Liu. 2014. Aspect and entity extraction for opinion mining. In *Data mining and knowledge discovery for big data*. Springer, 1–40.

[32] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. 2010. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 56–65.

[33] Yuan Zuo, Junjie Wu, Hui Zhang, Deqing Wang, Hao Lin, Fei Wang, and Ke Xu. 2015. Complementary Aspect-based Opinion Mining Across Asymmetric Collections. In *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 669–678.