Discovering Product Defects and Solutions from Online User Generated Contents

Xuan Zhang
Virginia Tech* & Sam's Club
*Blacksburg, Virginia
xuancs@vt.edu

Zhilei Qiao
University of Alabama at Birmingham
Birmingham, Alabama
qiaozl@uab.edu

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

Weiguo Fan
University of Iowa
Iowa City, Iowa
weiguo-fan@uiowa.edu

Edward A. Fox
Virginia Tech
Blacksburg, Virginia
fox@vt.edu

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

ABSTRACT

The recent increase in online user generated content (UGC) has led to the availability of a large number of posts about products and services. Often, these posts contain complaints that the consumers purchasing the products and services have. However, discovering and summarizing product defects and the related knowledge from large quantities of user posts is a difficult task. Traditional aspect opinion mining models, that aim to discover the product aspects and their corresponding opinions, are not sufficient to discover the product defect information from the user posts. In this paper, we propose Product Defect Latent Dirichlet Allocation model (PDLDA), a probabilistic model that identifies domain-specific knowledge about product issues using interdependent three-dimensional topics: Component, Symptom, and Resolution. A Gibbs sampling based inference method for PDLDA is also introduced. To evaluate our model, we introduce three novel product review datasets. Both qualitative and quantitative evaluations show that the proposed model results in apparent improvement in the quality of discovered product defect information. Our model has the potential to benefit customers, manufacturers, and policy makers, by automatically discovering product defects from online data.

CCS CONCEPTS

• **Information systems → Document topic models**; *Content analysis and feature selection*; • **Mathematics of computing →** *Bayesian computation.*

KEYWORDS

Product Defect Discovery; Topic Model; Opinion Mining

ACM Reference Format:
Xuan Zhang, Zhilei Qiao, Aman Ahuja, Weiguo Fan, Edward A. Fox, and Chandan K. Reddy. 2019. Discovering Product Defects and Solutions from Online User Generated Contents. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313732

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313732

1 INTRODUCTION

The number of Internet users almost tripled (4.2 Billion in Dec. 2018) in the past decade, and 3.4 Billion among them are social media users [1]. As a result, a vast volume of online UGC, including product defect related posts, is produced every day. For example, the official database of National Highway Traffic Safety Administration (NHTSA) has more than 1 million complaints related to the automobile industry. Even more posts are submitted to vehicle quality forums such as `toyotanation.com` and `honda-tech.com`. Large numbers of complaints occur in the customer service logs of manufacturers as well. This feedback can be useful for new customers to make purchase decisions, for existing customers to find resolutions to the problems, and for the manufacturers to improve the quality of products and services. Thus, identifying and summarizing product defect and resolution information from online user generated content has wide utility.

I have a 2002 Ford Explorer. **The o/d light was flashing and the transmission was slipping**. *We took it to our mechanic and it was diagnosed that the transmission needed to replace*	**I have an application completely stuck off screen. When I click on it main screen comes back but not the application!** *What app is it? try starting in safe mode (depress shift key when starting). That startup takes a while*.

Figure 1: Examples of product defect posts (bold: symptom, *italic*: resolution).

Two product defect posts from NHTSA and Apple MacBook forum, containing defect and solution, are shown in Figure 1. There are two important types of sentences in the posts: sentences in bold which describe the symptoms of the problem, and sentences in italics which illustrate resolution efforts to diagnose and fix the issue. In addition, there are component words, which may occur in both symptom and resolution sentences, and which give focus to a part, aspect, or subsystem of the product. Component, symptom, and resolution characterize product defects, and often occur as key entities in (or attributes of) product defect posts.

Domain-specific product defect mining can be treated as a natural extension of aspect opinion mining. Pioneered by the success of LDA-based models [3] in discovering latent representations of text,

[1]https://www.internetworldstats.com/emarketing.htm;
https://www.brandwatch.com/blog/amazing-social-media-statistics-and-facts/

most of the aspect-opinion mining research uses two-dimensional topic models following an "aspect + opinion" pattern, to summarize users' comments on a specific aspect [4, 11, 14, 15, 25]. In a sample post *The screen is very clear and great* [21], *screen* is an aspect, while *clear* and *great* are positive opinions.

The following three considerations explicate the distinctions between aspect opinion mining and product defect mining.

- *Component* (or *aspect*) is important in both types of mining; defects are always related to some property/feature of a product.
- The *opinion* entity is not sufficient to describe the other essential types of defect information. A product post usually contains information on *symptom* and *resolution*, which state how the defect manifests, and the fixes users/mechanics tried, respectively. Instead of *{aspect, opinion}*, an entity pattern of the form *{component, symptom, resolution}* is needed for modeling defects.
- The occurrence pattern of the key entities in consumer posts is distinctive. Traditional aspect opinion mining classifies text into "aspect" and "opinion" entities at the word level, which often occur as **word pairs** (e.g., "screen" and "clear" in the last example) [11]. But product defect mining needs to segment entities (especially "symptom" and "resolution") at the sentence level. Instead of occurring as word pairs, entities appear as **word groups** of varied sizes in posts.

Considering the unique characteristics of the product defect mining problem, we develop PDLDA, a three-dimensional LDA-based probabilistic model, which has the following primary contributions:

- The proposed model identifies the key entities of product defects – *component*, *symptom*, and *resolution* – from user posts as interdependent three-dimensional topics. For each entity of a defect, we have a corresponding topic. These topics are connected by the dependencies among entities.
- In contrast to the traditional aspect-opinion models, the proposed PDLDA model does not require word pairs. Existing LDA models for aspect summarization [10, 20, 22] usually take word pairs (e.g., an aspect word and an opinion word) as input. In contrast, PDLDA accepts user posts which have varying numbers of component, symptom, and resolution words, and even those without symptom or resolution entities.
- We share new datasets with practical utility that also can be employed in future studies.

2 PROPOSED MODEL

2.1 Problem Definition

Our work aims to build a model that can effectively identify three key entities of defects: *component*, *symptom*, and *resolution*. Component words (e.g., "engine", "brake", "power train") point out the flawed units which may cause the defect. Symptom words (e.g., "acceleration", "hesitate", "fail") describe what a product defect looks like. In contrast, resolution words (e.g., "check", "replace", "reset") show how people attempt to diagnose and fix the issues. Each of these entities is comprised of words, which usually follow a multinomial distribution across post documents. Furthermore, both symptom words and resolution words rely on the defective components. Therefore, our **research problem** is: given a dataset with D posts, the PDLDA model should produce a component topic φ for each of the K defects, which is the core entity. The component topic should be followed by symptom topics ψ and resolution topics τ which depend on it. Together, these joint topics should provide comprehensive information about a defect.

Table 1: Notation used in this paper.

K	number of topics
D	number of posts (documents)
V_c, V_s, V_r	size of component, symptom, and resolution word vocabularies, respectively
N_c, N_s, N_r	number of component, symptom, and resolution words in a post, respectively
C, S, R	Component, Symptom, and Resolution words in the corpus
H	stands for parameters of all Dirichlet distributions, including α, β, δ, γ, ϵ in the graphical diagram
φ, ψ, τ	word distributions (multinomial) over component, symptom, and resolution topics
θ, η, π	component topic distribution (multinomial) over documents, symptom topic distribution over component topics, and resolution topic distribution over component topics
λ_s, λ_r	Bernoulli distribution which decides whether a regular or a background topic is assigned to a word
t_s, t_r	indicator variables that decide whether a word is generated from a regular ($t = 0$) or a background topic ($t = 1$), for symptom and resolution words
z_i, y_i, x_i	topic assigned to the i^{th} component, symptom, or resolution word
$z^{\neg i}$,$y^{\neg i}$, $x^{\neg i}$	topic assignment vector for various types of words in the corpus excluding the i^{th} word
$n_c^T(d, k)$	number of component words in document d assigned to topic k; T represents topic distribution
$n_c^W(k, v_c)$	number of component words v_c assigned to component topic k; W represents word distribution
$n_s^T(k, l)$	number of symptom words assigned to topic l, whose corresponding component topic is k
$n_s^W(l, v_s)$	number of symptom words v_s assigned to symptom topic l
$n_r^T(k, m)$	number of resolution words assigned to topic m, whose corresponding component topic is k
$n_r^W(m, v_r)$	number of resolution words v_r assigned to resolution topic m
NTI_s, NTI_r	$normalized\ TF * IDF$ value of symptom or resolution word

2.2 Entity Extraction

When multiple types of entities are involved, we have to separate words of various types. In our case, we need to find the words belonging to component, symptom, and resolution entities, given a post. Some researchers [10, 20, 22, 24] classify words into categories before entering them into LDA models, while others assign a type to each word with the generative process [2, 16]. The first strategy is more efficient for product defect mining, since words in one sentence usually belong to the same entity (except that a component word may appear in symptom and resolution sentences). The entity extraction methods are introduced in Section 3.1.2 in detail.

2.3 PDLDA Model

As shown in Figure 2, PDLDA models key entities (component, symptom, and resolution) of defects with interdependent topics. There are three types of observed entities in the graphical model: component words (W_c), symptom words (W_s), and resolution words (W_r). The topics assigned to these words are z, y, and x, respectively. We use two conditional probability distributions η and π to model the dependence between symptom and resolution with the component. The notation used in our paper is described in Table 1. The post generation process is illustrated in Algorithm 1.

In the first five 1st-level loops, PDLDA determines the word distributions. For each component topic, a word distribution φ is

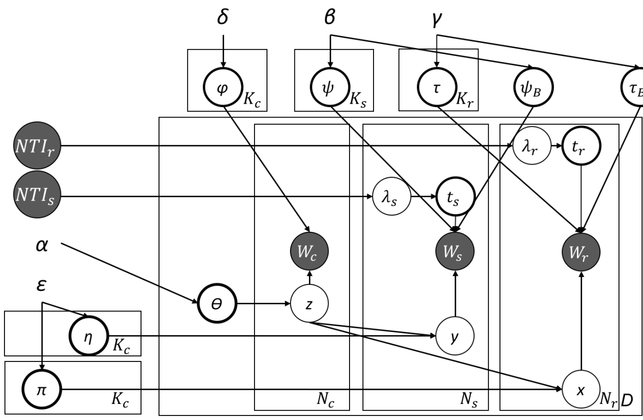


Figure 2: Plate notation of PDLDA model.

sampled from Dirichlet distribution δ. We also sample two corresponding symptom and resolution topic distributions, η and π, which depend on the component topic, from the Dirichlet distribution ϵ. They are used to create the dependence matrices. Word distributions ψ and τ for each symptom and resolution topic (both regular and background topics) are sampled, respectively.

The topic assignment is determined in the last 1st-level loop. Given a post d, we first sample a component topic z from Multinomial θ for each of the N_c component words in a document d. The component topic sampling is the same as with standard LDA. As a result, we get a set of topics, $Z_d = \{z_1, z_2, \cdots, z_{N_c}\}$. Next, to choose a y for each symptom word W_s, a component topic z' is randomly drawn from the component topic set Z_d, since we assume y depends on a certain z'. Then, we identify whether W_s is a *background* or *regular* symptom word by sampling an indicator t_s from a Bernoulli distribution λ_s. If the word is a regular symptom word ($t_s = 0$), we sample a symptom topic y from Multinomial $\eta_{z'}$ (a multinomial distribution assuming the current component topic is z') for each symptom word W_s, then sample the word from Multinomial ψ_y; otherwise ($t_s = 1$), the background symptom topic is assigned to the word, and we sample it from Multinomial ψ_B. The same sampling process is used during the generation of each resolution word W_r. Note PDLDA can be applied to a set of posts as long as they have a component entity; they do not have to have all three entities. PDLDA can sample topics for the existing entities even when a post has zero or one of: symptom or resolution entities.

A special feature of the PDLDA model is the presence of two switch variables t_s and t_r, which are designed to reduce overlap between different symptom (or resolution) topics. In particular, we give the most common words high probability of being assigned into a background symptom (or resolution) topic. Thus, a background topic will be formed by the words which are common to all the defects, such as “problem”, “work”, “stop”. The TF-IDF value of a word is used to decide whether it is a background or a regular word. Specifically, we draw indicators t_s and t_r from Bernoulli distributions λ_s and λ_r, respectively. The parameter of these distribution is initialized by the Normalized TF-IDF value (NTI_s or NTI_r) of that word, in particular, $P(t_s = 0) = normalized\ TF * IDF$.

Algorithm 1: Post generation process of PDLDA

for *each component topic* **do**
  Sample word distribution $\varphi \sim Dirichlet(\delta)$
  Sample conditional topic distribution $\eta \sim Dirichlet(\epsilon)$ for *symp* entities
  Sample conditional topic distribution $\pi \sim Dirichlet(\epsilon)$ for *reso* entities
for *each regular symptom topic* **do**
  Sample word distribution $\psi \sim Dirichlet(\beta)$
for *background symptom topic* **do**
  Sample word distribution $\psi_B \sim Dirichlet(\beta)$
for *each regular resolution topic* **do**
  Sample word distribution $\tau \sim Dirichlet(\gamma)$
for *background resolution topic* **do**
  Sample word distribution $\tau_B \sim Dirichlet(\gamma)$
for *each post d* **do**
  Sample *comp* topic distribution $\theta_d \sim Multinomial(\alpha)$
  for *each comp word W_c of d* **do**
    Sample a *comp* topic $z \sim Multinomial(\theta_d)$
    Sample $W_c \sim Multinomial(\varphi_z)$
  for *each symp word W_s of d* **do**
    Draw a *comp* topic z' from the set of z
    Sample an indicator $t_s \sim Bernoulli(\lambda_s)$
    if $t_s = 0$ **then**
      Sample a *symp* topic $y \sim Multinomial(\eta_{z'})$
      Sample $W_s \sim Multinomial(\psi_y)$
    else if $t_s = 1$ **then**
      Sample $W_s \sim Multinomial(\psi_B)$
  for *each reso word W_r of d* **do**
    Draw a *comp* topic z' from the set of z
    Sample an indicator $t_r \sim Bernoulli(\lambda_r)$
    if $t_r = 0$ **then**
      Sample a *reso* topic $x \sim Multinomial(\pi_{z'})$
      Sample $W_r \sim Multinomial(\tau_x)$
    else if $t_r = 1$ **then**
      Sample $W_r \sim Multinomial(\tau_B)$

Using this sampling strategy, especially sampling a component topic for related symptom and resolution topics, we eliminate a key drawback of the existing aspect mining LDA models [10, 20, 22] which require word pairs. For these methods, the opinion topic sampling for a opinion word relies on the aspect topic assigned to the aspect word in that word pair[2].

2.4 Model Inference

We will now provide the details of the inference of the proposed model which uses collapsed Gibbs sampling. The Gibbs updating rules discussed below show how PDLDA assigns topics to the component, symptom, and resolution words in a post.

The Gibbs updating rule for a component word i is:

$$P(z_i = k | z^{\neg i}, C, H) \propto \frac{n_c^T(d,k) + \alpha_k - 1}{\sum_{k'=1}^{K}(n_c^T(d,k') + \alpha_{k'}) - 1} \cdot \frac{n_c^W(k, v_c) + \delta_{v_c} - 1}{\sum_{v'=1}^{V_c}(n_c^W(k, v') + \delta_{v'}) - 1} \tag{1}$$

[2]This is the main reason why these models only accept word pairs as input.

Table 2: Datasets used in our evaluation.

Datasets	All posts	Labeled posts	Clusters
FOCUS (98-04)	10,099	1,941	4
MACBOOK	2,098	413	5
Patient.info	43,928	43,928	10

Assuming the corresponding component topic of symptom word i is k, the Gibbs updating rule for symptom words is :

$$P(y_i = l|z, y^{\neg i}, S, H) \propto \frac{n_s^T(k,l) + \epsilon_l - 1}{\sum_{l'=1}^{K}(n_s^T(k,l') + \epsilon_{l'}) - 1} \cdot \frac{n_s^W(l, v_s) + \beta_{v_s} - 1}{\sum_{v'=1}^{V_s}(n_s^W(l,v') + \beta_{v'}) - 1} \tag{2}$$

Assuming the corresponding component topic of resolution word i is k, the Gibbs updating rule for resolution words is:

$$P(x_i = m|z, x^{\neg i}, R, H) \propto \frac{n_r^T(k,m) + \epsilon_m - 1}{\sum_{m'=1}^{K}(n_r^T(k,m') + \epsilon_{m'}) - 1} \cdot \frac{n_r^W(m, v_r) + \gamma_{v_r} - 1}{\sum_{v'=1}^{V_r}(n_r^W(m,v') + \gamma_{v'}) - 1} \tag{3}$$

As mentioned before, the model decides whether a symptom (or resolution) word should be assigned with a background or regular symptom (or resolution) topic based on its TF-IDF value. The TF-IDF value is normalized for each post and used to set the parameter of the Bernoulli distribution λ_s or λ_r, from which we sample the type (background or regular) of that word.

3 EVALUATION

3.1 Experimental Setup

3.1.1 Datasets. To evaluate the usability of our model on different kinds of data, we use a wide range of datasets for our evaluation, including the complaint database of NHTSA[3], the problem discussion threads in the official forum of Apple[4], and the disease discussion threads in the Patient.info forum[5]. In the NHTSA and Apple datasets, we aim to discover the product defects and resolutions. In the Patient.info dataset, the human diseases can be considered analogous to product defects, and we aim to find their treatments, which is analogous to defect resolutions. Products which received the most complaints are selected from those data sources, and a post dataset is constructed for each, shown in Table 2. Specifically, we select posts of FORD FOCUS (1998-2004), Apple MacBook, and Patient.info to create 3 datasets. The entire datasets are used for topic coherence and qualitative evaluation. In addition, subsets of these datasets are manually labeled for post clustering evaluation. The post clusters are manually tagged by four Virginia Tech undergraduate students (each post tagged by 2 students), with an inter-annotator agreement rate of 81.2%. The post and cluster count of subsets are shown in columns 3 and 4 in Table 2.

[3]https://www-odi.nhtsa.dot.gov/downloads/flatfiles.cfm
[4]https://discussions.apple.com
[5]https://patient.info/forums

3.1.2 Data Preprocessing. Since this research primarily focusses more on the design of the topic model, rather than segmentation of the entities, we used light weight methods to implement entity segmentation. The frequent item set mining approach is used to extract components in all the datasets, by following the method introduced in [8]. Regarding the extraction of symptom and resolution entities, different strategies are used on various datasets.

- For the Apple forum and Patient.info datasets, we rely on the thread structure to separate them. Each forum thread is taken as a document. Then, we take the sentences in the first post as symptom sentences and those in the "most recommended solution" post as resolution sentences.
- A lexicon-based method is applied to the NHTSA datasets. Three types of sentences exist in the NHTSA posts: ownership, symptom, and resolution. We go through the sentences of a small dataset which has 400 posts of different vehicle models, find the feature words (e.g., "buy", "purchase", and "own") of ownership sentences, and find the feature words (e.g., "fix", "remedy", and "resolve") of resolution sentences. In this way, two lexicons are created, which will help decide the sentence type according to the lexicon word occurrence.

Topic models are quite sensitive to noise such as stop-words. Thus, a filter based on POS tagging further removes less-informative candidate symptom and resolution words. Based on our observations, we assume the most informative words are nouns, verbs, and adjectives. We apply the Stanford POS tagger [18] to symptom and resolution sentences, keeping only nouns, verbs, and adjectives. Then, we can provide observations (three entities for each post) to the PDLDA model.

3.1.3 Implementation and Parameter Setting. PDLDA is implemented in Java. We use symmetric priors for the Dirichlet distribution parameters $\alpha, \beta, \delta, \gamma$, and ϵ, and set them with empirically derived values $\alpha = 0.01$, $\beta = \gamma = \delta = \epsilon = 0.001$.

3.2 Performance Evaluation

3.2.1 Evaluation Criteria. Both quantitative and qualitative evaluations are conducted for the evaluation of the proposed PDLDA model.

- *Entity Clustering Accuracy*: Precision, Recall, and F-Measure [11] metrics are selected to measure entity (e.g., symptom and resolution) clustering performance. We follow the process introduced in [23] to calculate them.
- *Topic Coherence*: The Pointwise Mutual Information (PMI) metric [12] is chosen for topic coherence measurement.
- *Qualitative Evaluation*: The defects of different product models are identified from posts as joint topics.

3.2.2 Baseline Methods.

- *Standard LDA* [3]: After entity extraction using the methods in last section, we run standard LDA on the words of three entities separately to get three kinds of topics.
- *ILDA* [10]: We get the component (aspect) and the symptom (opinion) topics by taking symptom sentences as input, and get the component (aspect) and the resolution (opinion) topics from resolution sentences. A dependency parser [5] is used to extract aspect and opinion entities.

Table 3: Average Precision, Recall, and F-1 (%) of post entity clustering.

Dataset	**FORD FOCUS**						**APPLE MacBook**						**Patient.info**					
Entity	**Symptom**			**Resolution**			**Symptom**			**Resolution**			**Symptom**			**Resolution**		
Method	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
LDA	69.09	55.47	56.35	68.72	25.00	36.55	31.1	24.84	27.33	32.32	25.94	28.69	43.72	45.97	44.45	31.17	29.98	30.32
ILDA	69.58	37.74	47.52	64.86	32.31	39.02	30.31	28.27	29.65	30.79	23.13	25.11	40.89	30.80	34.36	31.45	19.90	23.17
ME-LDA	64.38	26.53	23.77	73.62	25.94	22.69	37.44	**33.84**	34.98	35.39	**31.21**	31.93	41.19	41.30	40.52	30.88	30.47	30.39
PDLDA	**88.82**	**57.08**	**59.24**	**91.36**	**33.34**	**43.51**	**57.7**	32.84	**35.29**	**49.62**	30.25	**33.69**	**53.82**	**43.82**	**44.52**	**36.96**	**30.56**	**31.15**

- *ME-LDA* [25]: We use the same input used in ILDA to produce topics by ME-LDA. Our own component lexicon and the opinion lexicon in [8] are used to label word types.

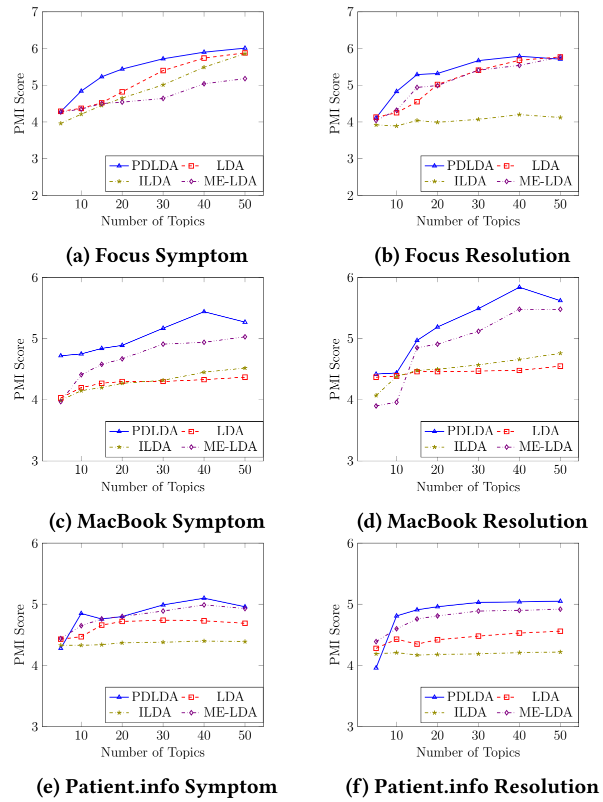
3.3 Experimental Results

3.3.1 Post Clustering. For this evaluation, the posts are clustered along two dimensions. First, they are clustered according to the symptom entity, then according to the resolution entity, using different models. When clustering symptom entities of posts, we take the symptom topic ID assigned to the majority of symptom words of a post as its symptom cluster. A similar clustering process occurs when the posts are clustered according to their resolution entities.

The clustering performance on the three datasets is shown in Table 3. We can see the precision and F-measure of PDLDA are better than the baselines, for both symptom and resolution clustering on all the datasets. ME-LDA has better recall on the MacBook dataset. Since PDLDA jointly models the three entities, it leverages the influence of component entities when clustering dependent entities. Therefore, it outperforms the standard LDA. In addition, PDLDA extracts more reasonable words on product defects than ILDA and ME-LDA; thus it captures more important features when clustering post entities. Also, the clustering performance on resolution entities is low for all of the models, compared to the symptom entities. This reveals that the words in resolution sentences are more difficult to separate. We observe that the words describing problem fixing measures are quite common, especially when we exclude the component words.

3.3.2 Topic Coherence. The topic coherence scores over 3 datasets are shown in Figure 3. Since the sampling of component topics in PDLDA is the same as the standard LDA, they perform quite similarly in terms of the topic coherence of components. Thus we only show the topic coherence of symptom and resolution topics. PDLDA outperforms the baseline methods in most cases, especially when the number of topics is over 15. Both the topic interdependency and entity identification mechanisms contribute to this. The PMI of PDLDA stops increasing and converges to the baselines when the topic number is very large. A possible reason is, the actual number of topics in the datasets is limited; thus, more overlap among topics occurs along with the topic number increase, which may prevent the PMI increase. ME-LDA outperforms LDA and ILDA, as it benefits from incorporating prior knowledge (i.e., tagged word types) when identifying word types. Simple models (LDA and ILDA) perform better when the number of topics is small (e.g., 5 topics).

3.3.3 Qualitative Evaluation. While capturing critical defects of a vehicle model, PDLDA uses joint topics, composed of the most probable component, symptom, and resolution words, to show the defect information. In our output, each defect is led by its



(a) Focus Symptom **(b) Focus Resolution** **(c) MacBook Symptom** **(d) MacBook Resolution** **(e) Patient.info Symptom** **(f) Patient.info Resolution**

Figure 3: PMI coherence of symptom and resolution topics.

component topic, followed by relevant symptom and resolution topics. The component topic distribution $p(\theta_{d,k})$ in document d is determined by Eq. (4). Given the component topic k, the most relevant symptom topic l can be obtained by sorting $\eta_{k,l}$, which is calculated by Eq. (5). Similarly, the most relevant resolution topic m given component topic k is obtained by sorting $\pi_{k,m}$, which is calculated by Eq. (6). The keywords of all types of topics can be figured out using the second part of the Gibbs updating rules.

$$\theta_{d,k} = \frac{n_c^T(d,k) + \alpha_k}{\sum_{k'=1}^{K}(n_c^T(d,k') + \alpha_{k'})} \tag{4}$$

$$\eta_{k,l} = \frac{n_s^T(k,l) + \epsilon_l}{\sum_{l'=1}^{K}(n_s^T(k,l') + \epsilon_{l'})} \tag{5}$$

$$\pi_{k,m} = \frac{n_r^T(k,m) + \epsilon_m}{\sum_{m'=1}^{K}(n_r^T(k,m') + \epsilon_{m'})} \tag{6}$$

Table 4: Example of topics extracted by LDA, ILDA, ME-LDA, and PDLDA (Nonrelevant words of each topic are underlined).

Component				Symptom				Resolution			
LDA	ILDA	ME-LDA	PDLDA	LDA	ILDA	ME-LDA	PDLDA	LDA	ILDA	ME-LDA	PDLDA
ignition	key	lock	ignition	turn	ford	turn	turn	dealer	have	replace	tow
key	focus	steer	key	lock	start	ignition	lock	contact	ford	ignition	lock
cylinder	car	remove	cylinder	start	turn	steer	key	repair	take	key	key
lock	vehicle	rotor	lock	steer	key	wheel	insert	maker	numerous	turn	locksmith
wheel	problem	open	wheel	key	put	lock	park	failure	recall	tow	call
				stick	get	start	stick	shop	resolve	start	drill
				unable	insert	key	remove	aware	receive	lock	common
				put	go	work	place	local	ignition	work	tumbler
				set	ignition	attempt	unlock	advise	get	call	advise
				unlock	common	stick	freeze	owner	defective	leave	aaa

Due to space constraints, Table 4 shows joint topics only for the "ignition" issue of FOCUS 98-04. The top 5 words are shown for component topics, while the top 10 words are shown for symptom/resolution topics. Looking at the joint topics produced by PDLDA, we see that the ignition component problem caused the key stuck in the lock so it could not turn, which was resolved by car towing and locksmith service. For each defect, users can locate the flawed units by reading the component topic, then learn what happens and how it is fixed, by reading the dependent topics. The nonrelevant words in the topics are underlined, through which we can see the topics generated by PDLDA have fewer nonrelevant and general words (e.g., "put" and "get" in column 6).

As demonstrated above, PDLDA can effectively and accurately identify defects and related solutions of products from online UGC, especially for products which people can fix by themselves, such as vehicles, electronics, etc. And it can also help patients find the symptoms and related treatments of diseases. Based on them, customers can make purchase decisions, manufacturers can improve products, governments can undertake administrative actions, and patients can find useful information for disease diagnosis and treatment.

4 RELATED WORK

Product defect mining is an extension of aspect-based opinion mining which has not been thoroughly investigated in the literature. Earlier works on product defect mining [1] classified product defects, as discussed in online forums, according to the level of severity. This method attempted to discover whether a defect exists in a review, or the severity of the defect, but did not account for the details of the defect (e.g. symptoms). Later, Tutubalina and Ivanov [19] extracted key phrases on product problems from user reviews according to the syntax dependencies among words. But, it relies on pre-defined syntax dependency rules, which is hard to generalize.

LDA based models have been successfully used for aspect-based opinion mining, and have incorporated various related factors, due to the unsupervised and flexible characteristics of LDA. Titov and McDonald [17] separated global and local opinion topics using a multiple-grain LDA model. Lin and He [9] pointed out the importance of sentiment in terms of opinion mining and added it as a key latent variable in their topic model. Zhao et al. [25] proposed a Maximum Entropy-LDA hybrid model to separate aspect words and aspect-specific opinion words. Ahuja et al. [2] incorporated Geotaggers into their aspect-opinion model by assuming people living in different places may discuss different topics and hold varied opinions. Park et al. [13] proposed an LDA model which connected Mobile Application (APP) descriptions with corresponding reviews in order to improve mobile APP searching. A holistic model which included most of the above key entities (e.g., aspect, opinion, sentiment, and granularity) was proposed by Wang et al. [21]. They further boosted their LDA model with domain knowledge learned by lifelong machine learning, which can be regarded as an extension of [7] and [6]. Rakesh et al. [16] suggested a spike-and-slab prior over the document-topic distribution, which generates fine-grained summaries of online reviews. However, the entities captured by these traditional aspect opinion mining models are not specific enough for product defect mining. In addition, these models cannot adapt to the characteristics of defect description (e.g., key entity locations).

5 CONCLUSION

This paper addresses the problem of product defect identification. To identify the key information of product defects, we develop a PDLDA model which extracts interdependent topics (component, symptom, and resolution) from UGC. This model differs from existing aspect summarization models as it jointly identifies the key entities of product defects as interdependent three-dimensional topics. In addition, PDLDA eliminates the dependence on word pairs that are required by the existing models. Collapsed Gibbs sampling is used for model inference. Both quantitative and qualitative evaluations are performed. PDLDA is shown to effectively identify product defects and to outperform existing methods. Since the (three) types of entities modeled by PDLDA frequently exist in the posts on different products, it can be easily generalized to health applications as well as other products, especially to products which people can fix by themselves, such as software, cell phones, cameras, etc. As future work, we plan to explore other entities related to product defects, such as user requirements and product improvements. The datasets and code will be released on Github[6].

Acknowledgments

This work was supported in part by the US National Science Foundation grants IIS-1619028, IIS-1707498, and IIS-1838730, and NVIDIA Corporation.

[6]https://github.com/aizest/PDLDA

REFERENCES

[1] Alan S Abrahams, Jian Jiao, G Alan Wang, and Weiguo Fan. 2012. Vehicle defect discovery from social media. *Decision Support Systems* 54, 1 (2012), 87–97.

[2] Aman Ahuja, Wei Wei, Wei Lu, Kathleen M Carley, and Chandan K Reddy. 2017. A Probabilistic Geographical Aspect-Opinion Model for Geo-tagged Microblogs. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 721–726.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of machine learning research* 3 (2003), 993–1022.

[4] 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.

[5] Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 740–750.

[6] 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.

[7] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Leveraging Multi-Domain Prior Knowledge in Topic Models.. In *IJCAI*, Vol. 13. 2071–77.

[8] 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.

[9] Chenghua Lin and Yulan He. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. ACM, 375–384.

[10] Samaneh Moghaddam and Martin Ester. 2011. ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 665–674.

[11] 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.

[12] 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.

[13] Dae Hoon Park, Mengwen Liu, ChengXiang Zhai, and Haohong Wang. 2015. Leveraging User Reviews to Improve Accuracy for Mobile APP Retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 533–542.

[14] Dae Hoon Park, C Zhai, and Lifan Guo. 2015. SpecLDA: Modeling product reviews and specifications to generate augmented specifications. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM.

[15] Michael Paul and Roxana Girju. 2010. A Two-dimensional Topic-aspect Model for Discovering Multi-faceted Topics. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI'10)*. AAAI Press, 545–550.

[16] 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 *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1573–1582.

[17] 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.

[18] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 173–180.

[19] Elena Tutubalina and Vladimir Ivanov. 2014. Unsupervised approach to extracting problem phrases from user reviews of products. *COLING 2014*. (2014), 48.

[20] Hao Wang and Martin Ester. 2014. A sentiment-aligned topic model for product aspect rating prediction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1192–1202.

[21] 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*. International World Wide Web Conferences Steering Committee, 167–176.

[22] Wei Xue, Tao Li, and Naphtali Rishe. 2015. Aspect and ratings inference with aspect ratings: supervised generative models for mining hotel reviews. In *Web Information Systems Engineering–WISE 2015*. Springer, 17–31.

[23] Mohammed J Zaki, Wagner Meira Jr, and Wagner Meira. 2014. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press.

[24] Tian-Jie Zhan and Chun-Hung Li. 2011. Semantic dependent word pairs generative model for fine-grained product feature mining. In *Advances in Knowledge Discovery and Data Mining*. Springer, 460–475.

[25] 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*. 56–65.