Interpreting the Public Sentiment Variations on Twitter

Shulong Tan, Yang Li, Huan Sun, Ziyu Guan* , Xifeng Yan, *Member, IEEE,* Jiajun Bu, *Member, IEEE,* Chun Chen, *Member, IEEE,* and Xiaofei He, *Member, IEEE*

Abstract— Millions of users share their opinions on Twitter, making it a valuable platform for tracking and analyzing public sentiment. Such tracking and analysis can provide critical information for decision making in various domains. Therefore it has attracted attention in both academia and industry. Previous research mainly focused on modeling and tracking public sentiment. In this work, we move one step further to interpret sentiment variations. We observed that emerging topics (named *foreground topics*) within the sentiment variation periods are highly related to the genuine reasons behind the variations. Based on this observation, we propose a Latent Dirichlet Allocation (LDA) based model, Foreground and Background LDA (FB-LDA), to distill foreground topics and filter out longstanding *background topics*. These foreground topics can give potential interpretations of the sentiment variations. To further enhance the readability of the mined reasons, we select the most representative tweets for foreground topics and develop another generative model called Reason Candidate and Background LDA (RCB-LDA) to rank them with respect to their "popularity" within the variation period. Experimental results show that our methods can effectively find foreground topics and rank reason candidates. The proposed models can also be applied to other tasks such as finding topic differences between two sets of documents.

Index Terms—Twitter, Public Sentiment, Emerging Topic Mining, Sentiment Analysis, Latent Dirichlet Allocation, Gibbs Sampling.

1 Introduction

With the explosive growth of user generated messages, Twitter has become a social site where millions of users can exchange their opinion. Sentiment analysis on Twitter data has provided an economical and effective way to expose public opinion timely, which is critical for decision making in various domains. For instance, a company can study the public sentiment in tweets to obtain users' feedback towards its products; while a politician can adjust his/her position with respect to the sentiment change of the public.

There have been a large number of research studies and industrial applications in the area of public sentiment tracking and modeling. Previous research like O'Connor *et al*. [19] focused on tracking public sentiment on Twitter and studying its correlation with consumer confidence and presidential job approval polls. Similar studies have been done for investigating the reflection of public sentiment on stock markets [4] and oil price indices [3]. They reported that events in real life indeed have a significant and immediate effect on the public sentiment on Twitter. However, none of these studies performed further analysis to mine useful insights behind significant sentiment variation, called *public sentiment variation*. One valuable analysis is to find possible reasons behind sentiment variation, which can provide important decision-making information. For example, if negative sentiment towards Barack Obama increases significantly, the White House Administration Office may be eager to know why people have changed their opinion and then react accordingly to reverse this trend. Another example is, if public sentiment changes greatly on some products, the related companies may want to know why their products receive such feedback.

It is generally difficult to find the exact causes of sentiment variations since they may involve complicated internal and external factors. We observed that the emerging topics discussed in the variation period could be highly related to the genuine reasons behind the variations. When people express their opinions, they often mention reasons (e.g., some specific events or topics) that support their current view. In this work, we consider these emerging events/topics as possible reasons.

Mining emerging events/topics is challenging: (1) The tweets collection in the variation period could be very noisy, covering irrelevant "background" topics which had been discussed for a long time and did not contribute to the changes of the public's opinion. How to filter out such background topics is an issue

* *Corresponding author*

- *S. Tan, J. Bu and C. Chen are with Zhejiang Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou, China, 310027. E-mail: {shulongtan, bjj, chenc}@zju.edu.cn*
- *Y. Li, H. Sun and X. Yan are with the Department of Computer Science, University of California, Santa Barbara, CA 93106. E-mail: {yangli, huansun, xyan}@cs.ucsb.edu*
- *Z. Guan is with the College of Information and Technology, Northwest University of China, Xi'an, CN 710127. E-mail: welbyhebei@gmail.com*
- *X. He is with State Key Laboratory of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, China, 310027. Email: xiaofeihe@cad.zju.edu.cn*

we need to solve. Text clustering and summarization techniques [5], [16] are not appropriate for this task since they will discover all topics in a text collection. (2) The events and topics related to opinion variation are hard to represent. Keywords produced by topic modeling [2] can describe the underlying events to some extent. But they are not as intuitive as natural language sentences. (3) Reasons could be complicated and involve a number of events. These events might not be equally important. Therefore, the mined events should be ranked with respect to their contributions.

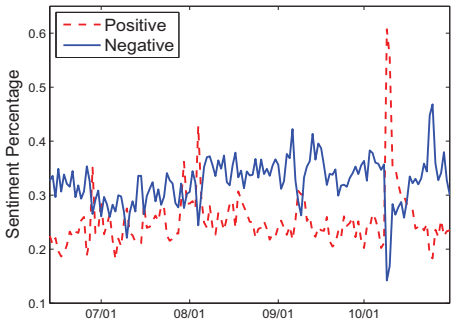


Fig. 1. Sentiment variation tracking of "Obama".

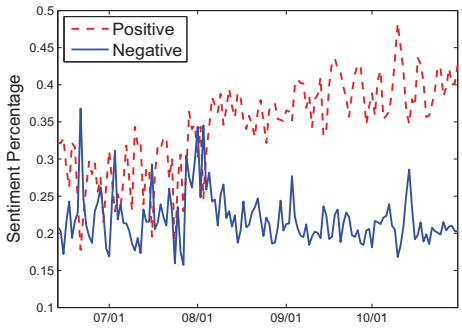


Fig. 2. Sentiment variation tracking of "Apple".

In this paper, we analyze public sentiment variations on Twitter and mine possible reasons behind such variations. To track public sentiment, we combine two state-of-the-art sentiment analysis tools to obtain sentiment information towards interested targets (e.g., "Obama") in each tweet. Based on the sentiment label obtained for each tweet, we can track the public sentiment regarding the corresponding target using some descriptive statistics (e.g., Sentiment Percentage). On the tracking curves significant sentiment variations can be detected with a pre-defined threshold (e.g., the percentage of negative tweets increases for more than 50%). Figures 1 and 2 depict the sentiment curves for "Obama" and "Apple." Note that in both figures, due to the existence of neutral sentiment, the sentiment percentages of positive and negative tweets do not necessarily sum to 1.

We propose two Latent Dirichlet Allocation (LDA) based models to analyze tweets in significant variation periods, and infer possible reasons for the variations. The first model, called Foreground and Background LDA (FB-LDA), can filter out background topics and extract foreground topics from tweets in the variation period, with the help of an auxiliary set of background tweets generated just before the variation. By removing the interference of longstanding background topics, FB-LDA can address the first aforementioned challenge. To handle the last two challenges, we propose another generative model called Reason Candidate and Background LDA (RCB-LDA). RCB-LDA first extracts representative tweets for the foreground topics (obtained from FB-LDA) as reason candidates. Then it will associate each remaining tweet in the variation period with one reason candidate and rank the reason candidates by the number of tweets associated with them. Experimental results on real Twitter data show that our method can outperform baseline methods and effectively mine desired information behind public sentiment variations.

In summary, the main contributions of this paper are two-folds: (1) To the best of our knowledge, our study is the first work that tries to analyze and interpret the public sentiment variations in microblogging services. (2) Two novel generative models are developed to solve the reason mining problem. The two proposed models are general: they can be applied to other tasks such as finding topic differences between two sets of documents.

2 Models for Sentiment Variation Analysis

To analyze public sentiment variations and mine possible reasons behind these variations, we propose two Latent Dirichlet Allocation (LDA) based models: (1) Foreground and Background LDA (FB-LDA) and (2) Reason Candidate and Background LDA (RCB-LDA). In this section we illustrate the intuitions and describe the details of the two proposed models.

2.1 Intuitions and Notations

As discussed in the last section, it is hard to find the exact causes for sentiment variation. However, it is possible to find clues via analyzing the relevant tweets within the variation period, since people often justify their opinion with supporting reasons. For example, if we want to know why positive sentiment on "Obama" increases, we can analyze the tweets with positive sentiment in the changing period and dig out the underlying events/topics co-occurring with these positive opinions.

We consider the emerging events or topics that are strongly correlated with sentiment variations, as possible reasons. Mining such events/topics is not trivial. Topics discussed before the variation period may continue receiving attention for a long time. Therefore, we need to make use of the tweets generated just

before the variation period to help "eliminate" these *background* topics. We can formulate this special topic mining problem as follows: *given two document sets, a background set B and a foreground set T, we want to mine the special topics inside T but outside B.* In our reason mining task, the foreground set T contains tweets appearing within the variation period and the background set B contains tweets appearing before the variation period. Note that this problem setting is general: it has applications beyond sentiment analysis.

To solve this topic mining problem, we develop a generative model called Foreground and Background LDA(FB-LDA). Figure 3 (a) shows the graphical structure of dependencies of FB-LDA. Benefiting from the reference role of the background tweets set, FB-LDA can distinguish the foreground topics out of the background or noise topics. Such foreground topics can help reveal possible reasons of the sentiment variations, in the form of word distributions. Details of FB-LDA will be described in Section 2.2.

FB-LDA utilizes word distributions to reveal possible reasons, which might not be easy for users to understand. Therefore we resort to finding representative tweets that reflect foreground topics learnt from FB-LDA. These most relevant tweets, defined as *Reason Candidates* C, are sentence-level representatives for foreground topics. Since they are not equally important, we rank these candidates (representative tweets) by associating the tweets in the foreground tweets set to them. Each tweet is mapped to only one candidate. The more important one reason candidate is, the more tweets it would be associated with. Top-ranked candidates will likely reveal the reasons behind sentiment variations.

In particular, the association task can be done by comparing the topic distributions (obatined by topic modeling methods, such as LDA) of tweets and the reason candidates. However, this solution is not optimal since the optimization goal of the topic modeling step does not take into account the tweet-candidate association at all. Inspired by [12], we propose another generative model called Reason Candidate and Background LDA (RCB-LDA) to accomplish this task. RCB-LDA can simultaneously optimize topic learning and tweet-candidate association. RCB-LDA, as depicted in Figure 3 (b), is an extension of FB-LDA. It will accept a set of reason candidates as input and output the associations between tweets and those reason candidates. Details of RCB-LDA will be described in Section 2.3.

For the purpose of better describing our models, we summarize all the notations used in FB-LDA and RCB-LDA in Table 1.

TABLE 1
Notations for our proposed models.

Symbols	Descriptions
C, T, B	the set of reason candidates, foreground tweets and background tweets
c, t, b	a candidate in C, or a tweet in T or B
$\mathbf{w}_c, \mathbf{w}_t, \mathbf{w}_b$	the word set in C, T or B
$\mathbf{w}'_t, \mathbf{w}''_t$	the word set in T with foreground topics or background topics, $\mathbf{w}_t = \{\mathbf{w}'_t, \mathbf{w}''_t\}$
N_c, N_t, N_b	the number of words in a candidate or a tweet
V	the number of words in the vocabulary
w_c^i, w_t^i, w_b^i	the ith word in c, t or b
θ, μ	the topic distribution of a tweet or a candidate
$\mathbf{z}_c, \mathbf{z}_t, \mathbf{z}_b$	the topic association set for words in $\mathbf{w}_c, \mathbf{w}_t$ or $\mathbf{w}_b$
$\mathbf{z}'_t, \mathbf{z}''_t$	the topic association set for words in $\mathbf{w}'_t$ or $\mathbf{w}''_t$, $\mathbf{z}_t = \{\mathbf{z}'_t, \mathbf{z}''_t\}$
z_c^i, z_t^i, z_b^i	the topic with which the ith word in c, t or b is associated
λ_t	the type decision distribution of t (choosing topics from foreground or background topics)
γ_t	the candidate association distribution of t (associated to which candidates)
$y_{t,i}$	the type associated with the ith word in t, $y_{t,i} =$ 0 or 1
$\mathbf{y}$	the type association set
$c_{t,i}$	the candidate with which the ith word in t is associated
$\mathbf{c}$	the candidate association set
K_f, K_b	the number of foreground or background topics
k_f, k_b	a topic index in foreground or background topics
$\phi_{f;k_f}$, $\phi_{b;k_b}$	the word distribution of foreground topic k_f or background topic k_b
α, β	the fixed parameters of Dirichlet priors

2.2 Foreground and Background LDA

To mine foreground topics, we need to filter out all topics existing in the background tweets set, known as background topics, from the foreground tweets set. we propose a generative model FB-LDA to achieve this goal.

As shown in Figure 3 (a), FB-LDA has two parts of word distributions: ϕ_f ($K_f \times V$) and ϕ_b ($K_b \times V$). ϕ_f is for foreground topics and ϕ_b is for background topics. K_f and K_b are the number of foreground topics and background topics, respectively. V is the dimension of the vocabulary. For the background tweets set, FB-LDA follows a similar generative process with the standard LDA [2]. Given the chosen topic, each word in a background tweet will be drawn from a word distribution corresponding to one background topic (i.e., one row of the matrix ϕ_b). However, for the foreground tweet set, each tweet has two topic distributions, a foreground topic distribution θ_t and a background topic distribution μ_t. For each word in a foreground tweet, an association indicator y_t^i, which is drawn from a type decision distribution λ_t, is required to indicate choosing a topic from θ_t or μ_t. If $y_t^i = 0$, the topic of the word will be drawn from foreground topics (i.e., from θ_t), as a result the word is drawn from ϕ_f based on the drawn topic. Otherwise ($y_t^i = 1$), the topic of the word will be drawn from background topics (i.e., from μ_t) and accordingly the word is drawn from ϕ_b.

With the help of background tweets, tweets coming from the foreground set but corresponding to background topics would make a bigger contribution

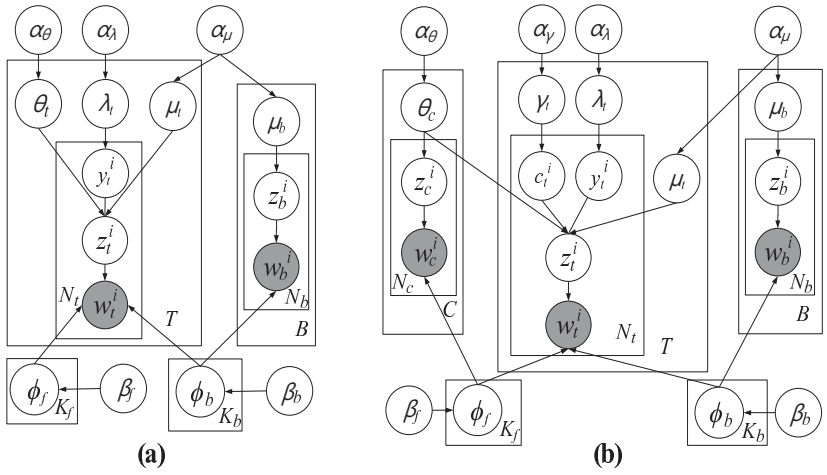


Fig. 3. (a) Foreground and Background LDA (FB-LDA); (b) Reason Candidate and Background LDA (RCB-LDA).

in background topics (i.e., ϕ_b) learning than in foreground topics (i.e., ϕ_f) learning. The large amount of similar background tweets in the background set would pull them to the background topics. Only tweets corresponding to foreground topics (i.e, emerging topics) will be used to build foreground topics. In this way, background topics will be filtered out and foreground topics will be highlighted in a natural way.

To summarize, we have the following generative process in FB-LDA:

1) Choose a word distribution $\phi_{f;k_f} \sim Dirichlet(\beta_f)$ for each foreground topic k_f.
2) Choose a word distribution $\phi_{b;k_b} \sim Dirichlet(\beta_b)$ for each background topic k_b.
3) For each tweet b in the background data, $b \in \{1, ..., B\}$:
   a) Choose a topic distribution $\mu_b \sim Dirichlet\ (\alpha_\mu)$.
   b) For each word w_b^i in the tweet, $i \in \{1, ..., N_b\}$:
      i) Choose a topic $z_b^i \sim Multinomial(\mu_b)$.
      ii) Choose a word $w_b^i \sim Multinomial(\phi_{b;z_b^i})$.
4) For each tweet t in the foreground data, $t \in \{1, ..., T\}$:
   a) Choose a type decision distribution $\lambda_t \sim Dirichlet(\alpha_\lambda)$.
   b) For each word w_t^i in the tweet, $i \in \{1, ..., N_t\}$:
      i) Choose a type $y_t^i \sim Bernoulli(\lambda_t)$.
      ii) if $y_t^i = 0$:
         A) Choose a foreground topic distribution $\theta_t \sim Dirichlet\ (\alpha_\theta)$.
         B) Choose a topic $z_t^i \sim Multinomial(\theta_t)$.
         C) Choose a word $w_t^i \sim Multinomial\ (\phi_{f;z_t^i})$.
      iii) else (i.e., $y_t^i = 1$.):
         A) Choose a background topic distribution $\mu_t \sim Dirichlet\ (\alpha_\mu)$.
         B) Choose a topic $z_t^i \sim Multinomial(\mu_t)$.
         C) Choose a word $w_t^i \sim Multinomial\ (\phi_{b;z_t^i})$.

Given the hyperparameters $\alpha_\theta, \alpha_\lambda, \alpha_\mu, \beta_f, \beta_b$, the joint distribution is:

$$\begin{aligned}\mathcal{L} &= P(\mathbf{y}, \mathbf{z}_t, \mathbf{z}_b, \mathbf{w}_t, \mathbf{w}_b | \alpha_\theta, \alpha_\lambda, \alpha_\mu, \beta_f, \beta_b) \\ &= P(\mathbf{y}|\alpha_\lambda) P(\mathbf{z}_t'|\mathbf{y}=0; \alpha_\theta) P(\mathbf{z}_t'', \mathbf{z}_b|\mathbf{y}=1; \alpha_\mu) \\ &\quad P(\mathbf{w}_t'|\mathbf{y}=0, \mathbf{z}_t'; \beta_f) P(\mathbf{w}_t'', \mathbf{w}_b|\mathbf{y}=1, \mathbf{z}_t'', \mathbf{z}_b; \beta_b).\end{aligned} \quad (1)$$

2.3 Reason Candidate and Background LDA

Different from FB-LDA, RCB-LDA needs a third document set: reason candidates. Reason candidates are in the form of natural language snippets and represent some specific events. In this paper we automatically select reason candidates by finding the most relevant tweets (i.e., representative tweets) for each foreground topic learnt from FB-LDA, using the following measure:

$$Relevance(t, k_f) = \sum_{i \in t} \phi_f^{k_f, i}, \quad (2)$$

where $\phi_f^{k_f}$ is the word distribution for the foreground topic k_f and i is the index of each non-repetitive word in tweet t. Note that we don't normalize this measure with respect to the length of the tweet, since tweets are all very short and generally have similar lengths. For other kinds of texts, normalization shall be applied. After filtering out junk tweets and merging similar ones, we consider the remaining relevant tweets as the reason candidates.

As shown in Figure 3 (b), the generative process of RCB-LDA is similar to that of FB-LDA. It generates the reason candidates set and the background tweets set in a similar way as the standard LDA. The main difference lies in the generative process of the foreground tweets set. Each word in the foreground tweets set can select a topic from alternative topic distributions: (1) draw a foreground topic from the

topic distribution of one candidate; (2) draw a background topic from its own background distribution μ_t. Specifically, for each word in tweets from the foreground tweets set, a y_t^i is chosen, similar to that in FB-LDA. If $y_t^i = 0$, we should choose an association candidate c_t^i, which is drawn from a candidate association distribution γ_t. Then we draw a foreground topic from $\theta_{c_t^i}$ for that word. The generative process for $y_t^i = 1$ is as same as that in FB-LDA.

The mapping from a foreground tweet t to any reason candidate or a background topic can be controlled by $\gamma_{t,c}$ and $\lambda_{t,0}$. If $\lambda_{t,0}$ is bigger than an empirical threshold (e.g., 0.5), the tweet will be mapped to the candidate c which corresponds to the largest $\gamma_{t,c}$ value; otherwise it will be mapped to the background topic.

Due to the space limit, we omit some parts of the generative process of RCB-LDA which are similar to those in FB-LDA. Here we just present the generative process for foreground tweets set in RCB-LDA:

For each tweet t in the foreground tweet data, $t \in \{1, ..., T\}$:

1) Choose a type decision distribution $\lambda_t \sim$ *Dirichlet*(α_λ).
2) Choose a candidate association distribution $\gamma_t \sim$ *Dirichlet*(α_γ).
3) For each word w_t^i in the tweet, $i \in \{1, ..., N_t\}$:
   a) Choose a type $y_t^i \sim$ *Bernoulli*(λ_t).
   b) if $y_t^i = 0$:
      i) Choose a candidate $c_t^i \sim$ *Multinomial* (γ_t).
      ii) Choose a topic $z_t^i \sim$ *Multinomial*$(\theta_{c_t^i})$.
      iii) Choose a word $w_t^i \sim$ *Multinomial* $(\phi_{f;z_t^i})$.
   c) else (i.e., $y_t^i = 1$.):
      i) Choose a topic distribution $\mu_t \sim$ *Dirichlet*(α_μ).
      ii) Choose a topic $z_t^i \sim$ *Multinomial*(μ_t).
      iii) Choose a word $w_t^i \sim$ *Multinomial*$(\phi_{b;z_t^i})$.

Given the hyperparameters $\alpha_\theta, \alpha_\gamma, \alpha_\lambda, \alpha_\mu, \beta_f, \beta_b$, the joint distribution is:

$$\begin{aligned}\mathcal{L} &= P(\mathbf{y}, \mathbf{c}, \mathbf{z}_c, \mathbf{z}_t, \mathbf{z}_b, \mathbf{w}_c, \mathbf{w}_t, \mathbf{w}_b | \alpha_\theta, \alpha_\gamma, \alpha_\lambda, \alpha_\mu, \beta_f, \beta_b) \\ &= P(\mathbf{y}|\alpha_\lambda) P(\mathbf{c}|\alpha_\gamma) P(\mathbf{z}_c, \mathbf{z}_t'|\mathbf{y} = 0, \mathbf{c}; \alpha_\theta) \\ &\quad P(\mathbf{z}_t'', \mathbf{z}_b|\mathbf{y} = 1; \alpha_\mu) P(\mathbf{w}_c, \mathbf{w}_t'|\mathbf{y} = 0, \mathbf{z}_c, \mathbf{z}_t'; \beta_f) \\ &\quad P(\mathbf{w}_t'', \mathbf{w}_b|\mathbf{y} = 1, \mathbf{z}_t'', \mathbf{z}_b; \beta_b). \end{aligned} \quad (3)$$

Note that FB-LDA and RCB-LDA are general models. They can be used in all cases where one wants to mine special topics/events in one text collection (i.e., the foreground text collection), but not in another text collection (i.e., the background text collection).

2.4 Gibbs Sampling

Similar to the original LDA model, exact inference for our model is intractable. Several approximate inference methods are available, such as variational inference [2], expectation propagation [17] and Gibbs Sampling [7], [9]. We use Gibbs Sampling here, since it is easy to extend and it has been proved to be quite effective in avoiding local optima.

The sampling methods for the two models are similar to each other. Due to the space limit, we only focus on the detailed inference of the relatively more complicated model RCB-LDA, and the inference of FB-LDA can be derived similarly. The sampling methods of $\mathbf{z}_c$, $\mathbf{z}_b$, $\mathbf{z}_t$, $\mathbf{c}$ and $\mathbf{y}$ in RCB-LDA are as follows:

$$\begin{aligned} &P(z_c^i = k_f | \mathbf{z}_c^{-(c,i)}, \mathbf{z}_t, \mathbf{z}_b, \mathbf{w}_c, \mathbf{w}_t, \mathbf{w}_b, \mathbf{c}, \mathbf{y}; \alpha_\theta, \alpha_\gamma, \\ &\alpha_\lambda, \alpha_\mu, \beta_f, \beta_b) \\ \propto\ &\frac{\Theta_{v,k_f} + \beta_f - 1}{\Theta_{(\cdot),k_f} + V \cdot \beta_f - 1} \times (\Phi_{c,k_f} + \Omega_{c,k_f} + \alpha_\theta - 1); \end{aligned} \quad (4)$$

where v is the word token in the vocabulary that has the same word symbol with word i. Θ_{v,k_f} is the number of times word token v being drawn from the k_fth foreground topic. $\Theta_{(\cdot),k_f}$ is the total number of word tokens drawn from the k_fth foreground topic. Φ_{c,k_f} is the number of words in candidate c which choose topic k_f. Ω_{c,k_f} is the number of words in the foreground tweets set which are associated with candidate c and choose topic k_f.

$$\begin{aligned} &P(z_b^i = k_b | \mathbf{z}_b^{-(b,i)}, \mathbf{z}_c, \mathbf{z}_t, \mathbf{w}_c, \mathbf{w}_t, \mathbf{w}_b, \mathbf{c}, \mathbf{y}; \alpha_\theta, \alpha_\gamma, \\ &\alpha_\lambda, \alpha_\mu, \beta_f, \beta_b) \\ \propto &\frac{\Delta_{v,k_b} + \beta_b - 1}{\Delta_{(\cdot),k_b} + V \cdot \beta_b - 1} \times (\Psi_{b,k_b} + \alpha_\mu - 1); \end{aligned} \quad (5)$$

where Δ_{v,k_b} is the number of times word tokens v being assigned to the k_bth background topic. $\Delta_{(\cdot),k_b}$ is the total number of word tokens assigned to the k_bth background topic. Ψ_{b,k_b} is the number of words in tweet b which choose topic k_b.

We sample z_t^i, y_t^i and c_t^i at the same time via *Blocked Gibbs Sampling*, using the following two equations:

$$\begin{aligned} &P(z_t^i = k_f, c_t^i = c, y_t^i = 0 | \mathbf{z}_t^{-(t,i)}, \mathbf{z}_c, \mathbf{z}_b, \mathbf{w}_c, \mathbf{w}_t, \\ &\mathbf{w}_b, \mathbf{c}^{-(t,i)}, \mathbf{y}^{-(t,i)}; \alpha_\theta, \alpha_\gamma, \alpha_\lambda, \alpha_\mu, \beta_f, \beta_b) \\ \propto &\frac{\Theta_{v,k_f} + \beta_f - 1}{\Theta_{(\cdot),k_f} + V \cdot \beta_f - 1} \times \frac{\Phi_{c,k_f} + \Omega_{c,k_f} + \alpha_\theta - 1}{\Phi_{c,(\cdot)} + \Omega_{c,(\cdot)} + K_f \cdot \alpha_\theta - 1} \\ &\times \frac{R_{t,c} + \alpha_\gamma - 1}{R_t + C \cdot \alpha_\gamma - 1} \times (M_{t,0} + \alpha_\lambda - 1); \end{aligned} \quad (6)$$

$$\begin{aligned} &P(z_t^i = k_b, y_t^i = 1 | \mathbf{z}_t^{-(t,i)}, \mathbf{z}_c, \mathbf{z}_b, \mathbf{w}_c, \mathbf{w}_t, \mathbf{w}_b, \mathbf{c}, \\ &\mathbf{y}^{-(t,i)}; \alpha_\theta, \alpha_\gamma, \alpha_\lambda, \alpha_\mu, \beta_f, \beta_b) \\ \propto &\frac{\Delta_{v,k_b} + \beta_b - 1}{\Delta_{(\cdot),k_b} + V \cdot \beta_b - 1} \times \frac{\Psi_{t,k_b} + \alpha_\mu - 1}{\Psi_{t,(\cdot)} + K_b \cdot \alpha_\mu - 1} \\ &\times (M_{t,1} + \alpha_\lambda - 1); \end{aligned} \quad (7)$$

where $M_{t,0}$ is the number of words generated by foreground topics in tweet t, $M_{t,1}$ is the number of words generated by background topics in tweet t, and

M_t is the total number of words in tweet t. $R_{t,c}$ is the number of words in tweet t which are generated by the topics in reason candidate c, while R_t is the total number of words in tweet t which are generated by the foreground topics.

2.5 Parameter Estimation

Given the sampled topics $\mathbf{z}_c, \mathbf{z}_t, \mathbf{z}_b$, type association $\mathbf{y}$ and candidate associations $\mathbf{c}$ as well as the inputs: αs, βs, $\mathbf{w}_c$, $\mathbf{w}_t$ and $\mathbf{w}_b$, we can estimate model parameters λ, γ, θ, μ, ϕ_f and ϕ_b for RCB-LDA as follows. Again, the detailed parameter estimation for FB-LDA is omitted here. In our experiments, we empirically set model hyperparameters as $\alpha_\theta = 0.1$, $\alpha_\gamma = 0.1$, $\alpha_\lambda = 0.5$, $\alpha_\mu = 0.1$, $\beta_f = 0.01$ and $\beta_b = 0.01$.

$$\lambda_{t,i} = \frac{M_{t,i} + \alpha_\lambda}{M_t + 2 \cdot \alpha_\lambda}; \gamma_{t,c} = \frac{R_{t,c} + \alpha_\gamma}{R_t + C \cdot \alpha_\gamma}; \quad (8)$$

where $i = 0$ or 1. λ_t serves as a threshold to determine whether the tweet is a foreground topic tweet (e.g., $\lambda_{t,0} > 0.5$) or a background tweet. γ_t can be used to determine which candidate the tweet is associated with (e.g., by choosing the biggest $\gamma_{t,c}$).

$$\theta_{c,k_f} = \frac{\Phi_{c,k_f} + \Omega_{c,k_f} + \alpha_\theta}{\Phi_{c,(\cdot)} + \Omega_{c,(\cdot)} + K_f \cdot \alpha_\theta}; \quad (9)$$

$$\mu_{i,k_b} = \frac{\Psi_{i,k_b} + \alpha_\mu}{\Psi_{i,(\cdot)} + K_b \cdot \alpha_\mu}; \quad (10)$$

$$\phi_f^{k_f,v} = \frac{\Theta_{v,k_f} + \beta_f}{\Theta_{(\cdot),k_f} + V \cdot \beta_f}; \phi_b^{k_b,v} = \frac{\Delta_{v,k_b} + \beta_b}{\Delta_{(\cdot),k_b} + V \cdot \beta_b}; \quad (11)$$

ϕ_f is the word distribution for each foreground topic; ϕ_b is the word distribution for each background topic.

3 Tracking Public Sentiment

In our work, sentiment tracking involves the following three steps. First, we extract tweets related to our interested targets (e.g., "Obama", "Apple" *etc*), and preprocess the extracted tweets to make them more appropriate for sentiment analysis. Second, we assign a sentiment label to each individual tweet by combining two state-of-the-art sentiment analysis tools [6], [29]. Finally, based on the sentiment labels obtained for each tweet, we track the sentiment variation regarding the corresponding target using some descriptive statistics. Details of these steps will be described in the following subsections.

3.1 Tweets Extraction and Preprocessing

To extract tweets related to the target, we go through the whole dataset and extract all the tweets which contain the keywords of the target.

Compared with regular text documents, tweets are generally less formal and often written in an ad hoc manner. Sentiment analysis tools applied on raw tweets often achieve very poor performance in most cases. Therefore, preprocessing techniques on tweets are necessary for obtaining satisfactory results on sentiment analysis:

(1) Slang words translation: Tweets often contain a lot of slang words (e.g., *lol*, *omg*). These words are usually important for sentiment analysis, but may not be included in sentiment lexicons. Since the sentiment analysis tool [29] we are going to use is based on sentiment lexicon, we convert these slang words into their standard forms using the Internet Slang Word Dictionary[1] and then add them to the tweets.

(2) Non-English tweets filtering: Since the sentiment analysis tools to be used only work for English texts, we remove all non-English tweets in advance. A tweet is considered as non-English if more than 20 percent of its words (after slang words translation) do not appear in the GNU Aspell English Dictionary[2].

(3) URL removal: A lot of users include URLs in their tweets. These URLs complicate the sentiment analysis process. We decide to remove them from tweets.

3.2 Sentiment Label Assignment

To assign sentiment labels for each tweet more confidently, we resort to two state-of-the-art sentiment analysis tools. One is the SentiStrength[3] tool [29]. This tool is based on the LIWC [27] sentiment lexicon. It works in the following way: first assign a sentiment score to each word in the text according to the sentiment lexicon; then choose the maximum positive score and the maximum negative score among those of all individual words in the text; compute the sum of the maximum positive score and the maximum negative score, denoted as FinalScore; finally, use the sign of FinalScore to indicate whether a tweet is positive, neutral or negative. The other sentiment analysis tool is TwitterSentiment[4]. TwtterSentiment is based on a Maximum Entropy classifier [6]. It uses automatically collected 160,000 tweets with emoticons as noisy labels to train the classifier. Then based on the classifier's outputs, it will assign the sentiment label (positive, neutral or negative) with the maximum probability as the sentiment label of a tweet.

Though these two tools are very popular, their performance on real datasets are not satisfactory because a large proportion of tweets still contain noises after preprocessing. We randomly picked 1,000 tweets and manually labeled them to test the overall accuracy of these two tools. It turns out that SentiStrength and TwitterSentiment achieve 62.3% and 57.2% accuracy on this testing dataset, respectively. By analyzing more cases outside the testing set, we found that

1. http://www.noslang.com
2. http://aspell.net
3. http://sentistrength.wlv.ac.uk
4. http://twittersentiment.appspot.com

TwitterSentiment is very inclined to misjudge a non-neutral tweet as neutral, while SentiStrength is highly likely to make a wrong judgement when FinalScore is close to 0. Therefore, we design the following strategy to combine the two tools:

(1) If both tools make the same judgement, adopt this judgement;

(2) If the judgement of one tool is neutral while that of the other is not, trust the non-neutral judgement;

(3) In the case where the two judgements conflict with each other (i.e., one positive and one negative), trust SentiStrength's judgement if the absolute value of FinalScore is larger than 1; otherwise, trust TwitterSentiment's judgement.

By utilizing the above heuristic strategy, the accuracy on the testing dataset is boosted to 69.7%, indicating the effectiveness of combining the two tools.

Note that the low accuracy of sentiment analysis would affect the final result of reason mining. For example, the mined possible reasons for negative sentiment variations may contain positive or neutral tweets. The low accuracy is mainly due to the fact that sentiment analysis techniques are still not very reliable on data with much noises, such as tweets. Fortunately, our work uses aggregated sentiment of multiple tweets. As long as the error caused by sentiment analysis is not significantly biased, the result is still useful, as demonstrated later by our case study. In practice, any sentiment analysis methods which would achieve better performance can be plugged into our algorithm and improve the performance of reason mining.

3.3 Sentiment Variation Tracking

After obtaining the sentiment labels of all extracted tweets about a target, we can track the sentiment variation using some descriptive statistics. Previous work on burst detection usually chooses the variation of the total number of tweets over time as an indicator. However, in this work, we are interested in analyzing the time period during which the overall positive (negative) sentiment climbs upward while the overall negative (positive) sentiment slides downward. In this case, the total number of tweets is not informative any more since the number of positive tweets and negative tweets may change consistently. Here we adopt the percentage of positive or negative tweets among all the extracted tweets as an indicator for tracking sentiment variation over time. Based on these descriptive statistics, sentiment variations can be found using various heuristics (e.g., the percentage of positive/negative tweets increases for more than 50%). Figures 1 and 2 show the sentiment curves regarding "Obama" and "Apple" from June 2009 to October 2009. We will test our proposed methods on sentiment variations of these two targets.

4 Experiments

In this section, we first present the experiments on a Twitter dataset for mining possible reasons of public sentiment variations. The results demonstrate that our models outperform baseline methods in finding foreground topics and ranking reason candidates. Then we apply our models on scientific article data and product review data. These applications show that our models can be used in any cases where we need to mine special topics or events from one text collection in comparison with another text collection.

4.1 Twitter Dataset

Our proposed models are tested on a Twitter dataset to analyze public sentiment variations. The dataset is obtained from the Stanford Network Analysis Platform [34]. It spans from June 11, 2009 to December 31, 2009 and contains around 476 million tweets. It covers around 20-30% of all public tweets published on Twitter during that time period. We do our experiments on a subset of the dataset, which spans from June 13, 2009 to October 31, 2009. In this work, we choose two targets to test our methods, "Obama" and "Apple". These two targets are chosen as representatives of the political sphere and the business field, where the analysis of sentiment variation plays a critical role in helping decision making. Figures 1 and 2 show the sentiment variation curves of these two targets.

To evaluate our models, we choose 50 sentiment variations for "Obama" and "Apple", from June 13, 2009 to October 31, 2009. In particular, we first detect sentiment variation periods by the following heuristic: if the percentage of negative/postive tweets increases for more than 50%, we mark the increasing period as a sentiment variation period. Then the foreground tweets set is generated by tweets within the variation period, while the background tweets set is formed by tweets just before the variation period. The amount of background tweets is chosen to be twice as that of the foreground set. Both foreground set and background set only contain the tweets whose sentiment labels correspond to the sentiment variation (e.g., positive label for positive sentiment increase). Table 2 shows some basic statistics for the 50 cases.

	"Obama"	"Apple"
# Negative cases	14	11
# Positive cases	12	13

TABLE 2
Basic statistics for the 50 sentiment variations.

4.2 Foreground Topics from FB-LDA

In this section, we evaluate the effectiveness of our first proposed model FB-LDA, using all 50 sentiment variations shown in Table 2. We will first compare the learned topics of FB-LDA and LDA with a heuristic evaluation metric. Then we will devise two baseline

methods and quantitatively compare FB-LDA with them by using manually labeled ground truth.

4.2.1 Average Word Entropy

We first evaluate FB-LDA by measuring the quality of the learned foreground topics. Here we treat the standard LDA (only using the foreground tweets set) as the baseline. We propose to measure the quality of a topic using *word entropy*: the conditional entropy of the word distribution given a topic, which is similar to the topic entropy [8]. Entropy measures the average amount of information expressed by each assignment to a random variable. If the a topic's word distribution has low word entropy, it means that topic has a narrow focus on a set of words. Therefore, a topic modeling method with a low average word entropy generates topics with high clarity and interpretability. The definition of the word entropy given a topic k is as follows:

$$\begin{aligned} H(\mathbf{w}|k) &= -\sum_{i=1}^{V} \hat{p}(w_i|k)\texttt{log}\hat{p}(w_i|k) \\ &= -\sum_{i=1}^{V} \phi_{k,i}\texttt{log}\phi_{k,i}, \end{aligned} \tag{12}$$

where ϕ_k is the word distribution of topic k and V is the number of words in the vocabulary.

Here we configure FB-LDA to output 20 foreground topics and 20 background topics, and set LDA to produce 20 topics. The average word entropies for topics learned from FB-LDA and LDA are **3.775** and **4.389**, respectively. It shows that the topics produced by FB-LDA exhibit lower word entropy than those learned by LDA, indicating that FB-LDA can generally obtain topics with less ambiguity and more interpretability.

4.2.2 Quantified Evaluation

To verify that FB-LDA is effective in finding foreground topics, we first manually find foreground events for each sentiment variation and consider these foreground events as ground truth. Then we evaluate the precision and recall of the results mined by FB-LDA with respect to the ground truth.

To find ground truth we will first do text clustering on the foreground tweets set and background tweets set respectively. Big clusters in the foreground set will be extracted as candidates. Then we manually filter out the candidates which also exist in the background set. The remaining candidates will be treated as the final ground truth. Each candidate will be represented by 1-3 representative tweets which are different descriptions of the underlying event. On average, each sentiment variation has 9 emerging/foreground events.

The precision and recall of the results found by FB-LDA are computed as follows: (a) rank foreground topics by their word entropies in ascending order. (b) for a foreground topic, five most relevant tweets are selected by Equation (2). (c) if the most relevant tweets contain a tweet in the ground truth, or contain a tweet which is very similar to a tweet in the ground truth, we believe that the method finds a correct foreground event. We define the similarity between two tweets as follows:

$$Similarity(t_i, t_j) = \frac{|WordOverlap| * 2}{|t_i| + |t_j|}. \tag{13}$$

In our experiments, two tweets will be considered "similar" if their *Similarity* is no less than 0.8.

Since there is no existing work that does exactly the same thing as our work (i.e., foreground topics/events finding), we design two methods based on traditional techniques as the baselines: **k-means** and **LDA**. For k-means, we first run the k-means clustering on the foreground set and the background set respectively. Since clusters from the foreground set contain both foreground and background topics, we design a mechanism to filter out background clusters by comparing clusters between the foreground set and the background set. If a cluster corresponds to the same topic/event with one background cluster, it will be filtered out. In particular, we compute the cosine similarity of the two cluster centers. If the cosine similarity is bigger than a threshold s, we consider the two clusters as from the same topic. In this experiment, we empirically set $s = 0.8$ to achieve the best performance. After background topics filtering, the remaining foreground clusters will be ranked by their sizes in descending order. Then for each cluster we find five tweets which are closest to the cluster center. The evaluation method for k-means is as same as that of FB-LDA. For the second baseline LDA, the background topics filtering step is similar to k-means. But instead of comparing cluster centers, here we compare the word distributions of topics. For the threshold setting, we empirically set $s = 0.9$ to achieve the best performance. The topic ranking method and the evaluation step of LDA are the same as those of FB-LDA.

We observed that the most relevant tweets of each topic/cluster are similar with each other and could clearly represent the semantics of the topic/cluster. Moreover, each of the most relevant tweets generally corresponds to a specific event. Therefore, if a representative tweet of a foreground topic/cluster appears in the ground truth set, we could reasonably conclude that the foreground topic/cluster corresponds to one ground truth event.

Experimental results Figure 4 shows the Precision-Recall curves (average on all 50 variations) for FB-LDA, LDA and k-means. In this experiment, we configure FB-LDA to produce 20 foreground topics and 20 background topics. For LDA, we configure it to produce 20 topics on the foreground set and

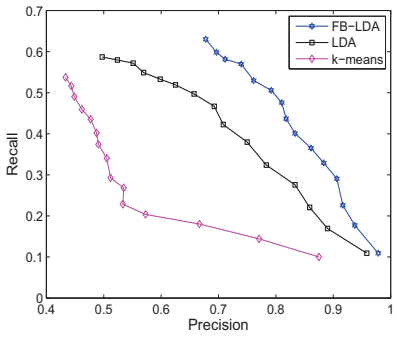


Fig. 4. Precision-Recall curves for FB-LDA, LDA and k-means.

another 20 topics on the background set. For k-means clustering, we run it on the two sets respectively, each generating 20 clusters. It can be found that FB-LDA greatly outperforms the two baselines in terms of precision and recall. LDA and k-means can not work well because a fixed threshold for filtering out background topics is obviously not appropriate for all cases. In comparison, FB-LDA can work properly without depending on any manually set thresholds.

4.3 Reason Ranking of RCB-LDA

In this section, we evaluate our second model RCB-LDA in ranking reason candidates.

4.3.1 Association Accuracy

We randomly choose five sentiment variations from all 50 cases in Table 2, two for "Obama" and three for "Apple." For each selected case, several reason candidates are generated (see Section 2.3). Then RCB-LDA ranks these candidates by assigning each tweet in the foreground tweets set to one of them or the background. Candidates associated with more tweets are more likely to be the main reasons. Before showing the reason ranking results, we first measure RCB-LDA's association accuracy and compare it with two baseline methods.

We manually label a subset of tweets in foreground set as the ground truth. Each label contains two elements: one tweet and one candidate (or the background). For each case, 1,000 tweets are manually labeled. Then we extend the labeled set by comparing labeled tweets' contents with the unlabeled tweets. If an unlabeled tweet has the same content with a labeled tweet, it should inherit the label from the labeled one.

Our model is compared with two baselines: (1) **TFIDF**: In this method, each tweet or candidate is represented as a vector with each component weighted by term frequency/inverse document frequency (TF-IDF). The association is judged by the cosine similarity between the two vectors representing tweets and candidates. (2) **LDA**: For this method, we use the standard LDA model to learn topics from foreground tweets and candidates together (i.e., treating the foreground tweets and reason candidates as an entire text set). Then we compute topic distribution distances between tweets and candidates using cosine similarity. We also tried to use the standard Jensen-Shannon(JS) divergence to measure similarity. But it turns out that the baseline methods perform even worse under this measure.

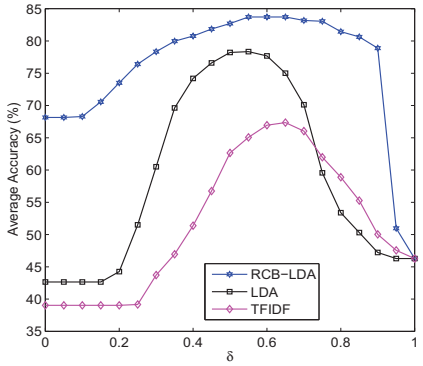


Fig. 5. Average association accuracy comparison for TFIDF, LDA and RCB-LDA by varying δ.

For all the three models, we evaluate them by measuring their mapping accuracy in assigning tweets to candidates/background. The mapping accuracy is defined as: *the number of correctly mapped tweets over the total number of tweets in the testing set*. The mapping is controlled by a threshold δ (indicating mapping to any candidate or to the background): (1) For TFIDF and LDA, a tweet is mapped to a candidate if there is at least one candidate whose relative similarity with the tweet is bigger than δ, and the candidate with the maximum similarity will be selected as the mapping destination; otherwise it is mapped to the background. The relative similarity between a tweet and a candidate is defined as: *the ratio of the similarity between them over the sum of similarities between the tweet and all candidates*. (2) For RCB-LDA, a tweet is mapped to the candidate corresponding to the largest $\gamma_{t,c}$ value if $\lambda_{t,0} > \delta$ and mapped to the background otherwise. The meanings of $\lambda_{t,0}$ and $\gamma_{t,c}$ can be found in Section 2.

Figure 5 shows the comparison of all three models' average mapping accuracies by varying δ. Our RCB-LDA model achieves the best accuracy in a wide range of the parameter variation. Moreover, compared with the two baseline methods, our method is not very sensitive to the varying threshold. LDA cannot work well for two reasons: (1) the topics learnt by LDA cannot accurately reflect the real foreground events; (2) LDA does not optimize the association goal directly.

4.3.2 Reason Ranking Example

Finally, we show an example of the reason ranking results. The example is a negative sentiment variation towards "Apple" from July 1st to July 3rd. Table 3

Cnt	Reasons
275	BREAKING Shooting at Arlington Apple Store! News Video via mashable. WTF.
191	Apple Patching Serious SMS Vulnerability on IPhone. Apple is Working to Fix an iPhone.
179	Apple warns on iPhone 3GS overheating risk.
101	Apple may drop NVIDIA chips in Macs following contract fight.
87	Child Porn Is Apple's Latest iPhone Headache.
84	App Store Rejections: Apple rejects iKaraoke app then files patent for karaoke player.

TABLE 3
Ranking results of reason candidates by RCB-LDA. This is an example of a negative sentiment variation towards "Apple" from July 1st to July 3rd.

shows the results. Here we set $\delta = 0.6$ and rank all the reason candidates with respect to the number of tweets associated with them.

As can be seen, the mined reasons are meaningful and reasonable. They clearly explain why people hold negative opinions during this period. The "shooting event at Apple store" is likely the main reason. Besides, "SMS vulnerability on iPhone" and "the iPhone 3GS overheating risk" are also important reason candidates. Furthermore, the possible reasons are represented as natural language sentences, which greatly eases the understanding of these reasons. Reasons shown in Table 3 might be "shallow" to informed observers, however, for people with little background knowledge, these results are very useful for showing them a big picture about the events behind the sentiment variation. Compared to professional analysis provided by news or TV editors, our models can give quantitative results and require much less manual efforts. Moreover, professional analysts can hardly be familiar with all the events regarding various targets (well known or not well known). Our methods can work on any target, in any time period. The mined results from our models can be used as references by those professional analysts.

4.4 Scientific Articles

As explained in Section 2.3, the proposed models are general in the sense that they can be used to mine special topics existing in one text collection, but not in another one. The task in this test is to find new topics of the information retrieval (IR) domain in recent three years, using FB-LDA. We expect to find research topics that emerge recently in IR domain, such as *Twitter Data Analysis* and *Hashing Techniques*. Long standing IR topics, such as *Retrieval Models* and *Web Page Analysis*, are not desired. Specifically, we plan to mine special topics in papers published in the proceedings of ACM SIGIR 2010, 2011 and 2012. The background set contains papers published in the SIGIR conference during 2000-2009. These papers are collected from the ACM Digital Library [5]. The foreground set contains 294 papers and the background set contains 630 papers. We only use the title and the abstract for each paper in this experiment.

5. http://dl.acm.org

In this test, we set FB-LDA to output 20 foreground topics along with 20 background topics, and set LDA to produce 20 topics. The average word entropy for special topics learnt from FB-LDA is **3.856** and that for topics from LDA is **4.275**. It shows that FB-LDA outperforms LDA in finding meaningful topics.

Tables 4 and 5 show the top 10 topics (foreground topics for FB-LDA) with the lowest word entropy learnt from FB-LDA and LDA, respectively. The second column of each table presents research topics by manual inspection. And the third column shows top words for each topics. As illustrated, most foreground topics found by FB-LDA are new and emerging in recent years. While many of the topics learnt from LDA are general IR topics which have been studied for a long time. It is clear that FB-LDA is superior in finding special/emerging topics.

4.5 Product Reviews

We then verify the effectiveness of RCB-LDA on product review data. The task in this experiment is to find aspects or features in which customers think Kindle 4 outperforms Kindle 3. Aspects being frequently talked in both Kindle 3 and 4 reviews, and aspects hardly being mentioned in Kindle 4 reviews should be ranked lower. In contrast, aspects being frequently discussed in Kindle 4 reviews but not in Kindle 3 reviews should be ranked higher.

In this test, we consider positive reviews for Kindle 4 as foreground documents and treat positive reviews for Kindle 3 as background documents. In order to test RCB-LDA, we need some reason/aspect candidates. These aspect candidates are automatically generated from the product descriptions, which are publicly available online. Specifically, we crawled 1,000 positive (5-star) reviews for Kindle 3 and Kindle 4 respectively, from *Amazon.com*. From the product descriptions we generate 9 candidates to describe different aspects of Kindle (e.g., Weight, Size *etc*). Each candidate contains around 5 sentences which are all about the same aspect.

Figure 6 shows the comparison of RCB-LDA results under two settings: (1) RCB-LDA with both foreground and background data; (2) RCB-LDA using only foreground data. In both tables, the first column shows the count of reviews assigned to each candidate by our model. Since reviews are usually much longer

	Research Topics	Top Words
1	Exploiting Users' Behavior Data	Behavior Search Model User Click Log Session Data
2	Probabilistic Factor Models for Recommendation	User Recommendation Person Interest Facet Factor Latent
3	Search Result Diversification	Result Search Vertical Diverse Diversify Subtopic Show
4	Query Suggestions	Query Search Suggest Engine Log Reformulation Predictor
5	Quality of User-generated Content	Label Quality Book Crowdsource Select Flaw Impact Sample
6	Twitter Stream Mining	Stream Twitter Context Tweet Entity Toponym Context-aware
7	Image Search and Annotation	Image Visual Attribute Estimate Face Privacy Flickr Facial
8	Search Result Cache Invalidation	Time Result Temporal Cache Evaluate Update Invalidate
9	Temporal Indexing	Collect Index Time Web Structure Temporal Archive Time-travel
10	Hashing for Scalable Image Retrieval	Retrieval Hash Example Code Method Propose Application

TABLE 4
Foreground topics learnt from FB-LDA in the task of finding new IR topics

	Research Topics	Top Words
1	Query Suggestion	Query Search Engine Result Suggest Relevant Web User Log Rank Select
2	Web Page Analysis	Web Page Text Ad Anchor Site Advertising Search Detect Spam External
3	Question and Answer Systems	Answer Question User Structure CQA Community Thread Vote Relationship
4	Personalized Recommendation	Recommendation User Item Personalized Rating Profile Filtering Collaborative
5	Retrieval Models	Document Model Retrieval Relevant Distribution List Score Single Ranking
6	Pseudo-Relevance Feedback	Query Term Word Language Retrieval Translate Feedback Pseudo-Relevance
7	Multimedia Retrieval	Feature Predict Content Learning Classify Extract Framework Music
8	Machine Learning in IR	Label Data Training Learning Domain Rank Select Classify Adapt Sample Active
9	Exploiting Users' Behavior Data	Search User Task Behavior Study Interaction Web Session Click Information
10	Social Media Analysis	Social User Network Information Tag Influence Interest Factor Relevant System

TABLE 5
Topics learnt from LDA in the task of finding new IR topics.

RCB-LDA w/o BG	
Cnt	*Aspects*
214	Special Offer
193	Weight
192	Size
166	Button
159	Price
153	Navigation
150	Screen
146	Sensibility
141	Storage

RCB-LDA	
Cnt	*Aspects*
200	Weight
190	Size
183	Button
133	Sensibility
129	Special Offer
122	Storage
93	Screen
90	Navigation
89	Price

Fig. 6. Experiment results on kindle reviews. The left table shows the result of running RCB-LDA using only Kindle 4 reviews (foreground data) and the right table shows the result of running RCB-LDA using both Kindle 4 reviews (foreground data) and Kindle 3 reviews (background data).

than tweets, each review can cover multiple aspects. Therefore we allow one review to be assigned to multiple candidates as long as $\gamma_{t,c}$ is bigger than a threshold ($\gamma_{t,c} > 0.2$ in this experiment). Note that in the experiments conducted on the Twitter data, one tweet is assigned to only one reason candidate.

As illustrated in Figure 6, without considering the background, the model will rank some aspects shared by both Kindle 3 and Kindle 4 (e.g., Special Offer, Price) at the top positions (the left table in Figure 6). This is not good since we are interested in detecting the aspects that customers think Kindle 4 outperforms Kindle 3. With the help of the background data, the results of RCB-LDA model can perfectly satisfy our demand. As shown by the right table, when taking the background into account, the RCB-LDA model will boost the rankings of aspects which clearly show the advantages of Kindle 4 over Kindle 3 (e.g., Weight, Size, Sensibility), and at the same time lower the rankings of shared aspects (e.g., Special Offer, Price).

The above two experiments demonstrate that our models are general and not limited to the possible reason mining problem. They can be applied to various tasks involving finding the topic differences between two sets of documents.

5 Related Work

To the best of our knowledge, this is the first work to analyze and interpret the public sentiment variations in microblogging services. Although there is almost no previous work on exactly the same problem, here we provide a brief review of related work from several greater perspectives.

Sentiment Analysis. In recent years, sentiment analysis, also known as opinion mining, has been widely applied to various document types, such as movie or product reviews[36], [11], webpages and blogs [35]. Pang *et al.* conducted a detailed survey of the existing methods on sentiment analysis [20]. As one main application of sentiment analysis, sentiment classification [32], [13] aims at classifying a given text to one or more pre-defined sentiment categories.

Online public sentiment analysis is an increasingly popular topic in social network related research. There have been some research work focusing on assessing the relations between online public sentiment and real-life events (e.g., consumer confidence, stock market [19], [3], [28]). They reported that events in real

life indeed have a significant and immediate effect on the public sentiment in Twitter. Based on such correlations, some other work [18], [31] made use of the sentiment signals in blogs and tweets to predict movie sales and elections. Their results showed that online public sentiment is indeed a good indicator for movie sales and elections. Different from the existing work in this line, we propose to analyze possible reasons behind the public sentiment variations.

Event Detection and Tracking. In this paper, we are interested in analyzing the latent reasons behind the public sentiment variations regarding a certain target. Considering the target as a given query, we can obtain all tweets about this target. Then there are two approaches to accomplish the task: (1) detecting and tracking events, and performing sentiment analysis for the tweets about each event; (2) tracking sentiment variations about the target and finding reasons for the sentiment changes.

The first approach is intuitive but problematic because detecting and tracking events are not easy, especially for fine-grained ones, such as one paragraph in some speech or one feature of a product. These events should be good reason candidates to explain the sentiment variation. However, we cannot detect and track them by existing event tracking methods [21], [15], [22], [1], [33], [10] which are only applicable to popular events, such as the biggest events per day in the whole Twitter message stream. Leskovec *et al.* proposed to track memes, such as quoted phrases and sentences, in the news cycle [14]. It is based on the quotation relations of memes. However, such quotation relations do not frequently appear in Twitter because of the length limit of tweets. "Retweets" in Twitter are different from quotations since retweets can only be used to track reposted messages with exactly the same content, but not to track relayed messages using different descriptions.

We choose the second approach, which first tracks sentiment variations and then detects latent reasons behind them. Through analyzing the output of our proposed models, we find that the second approach is able to find fine-grained reasons which correspond to specific aspects of events. These fine-grained events can hardly be detected by existing event detection and tracking methods.

Data Visualization. The reason mining task is committed to show specific information hidden in the text data. So it is also related to data visualization techniques. Recently, there are many excellent works which fucus on data visualization using subspace learning algorithms [24], [23], [25]. Unfortunately these works are not appropriate for text data especially for noisy text data. Another popular data visualization technique is ranking [26], [30]. Ranking is a core technique in the information retrieval domain which can help find the most relevant information for given queries. However, the reason mining task cannot be solved by ranking methods because there are no explicit queries in this task.

Correlation between Tweets and Events. To better understand events, there have been some work[21], [15], [33], [5], [16] trying to characterize events using tweets. However, few of them proposed to study the correlation between tweets and events. Recently, Hu *et al.* proposed novel models to map tweets to each segmentation in a public speech [12]. Our RCB-LDA model also focuses on finding the correlation between tweets and events (i.e, reason candidates). However, it is different from previous work in the following two aspects: (1) In our model, we utilize a background tweets set as a reference to remove noises and background topics. In this way, the interference of noises and background topics can be eliminated. (2) Our model is much more general. It can not only analyze the content in a single speech, but also handle more complex cases where multiple events mix together.

6 Conclusions

In this paper, we investigated the problem of analyzing public sentiment variations and finding the possible reasons causing these variations. To solve the problem, we proposed two Latent Dirichlet Allocation (LDA) based models, Foreground and Background LDA (FB-LDA) and Reason Candidate and Background LDA (RCB-LDA). The FB-LDA model can filter out background topics and then extract foreground topics to reveal possible reasons. To give a more intuitive representation, the RCB-LDA model can rank a set of reason candidates expressed in natural language to provide sentence-level reasons. Our proposed models were evaluated on real Twitter data. Experimental results showed that our models can mine possible reasons behind sentiment variations. Moreover, the proposed models are general: they can be used to discover special topics or aspects in one text collection in comparison with another background text collection.

Acknowledgments

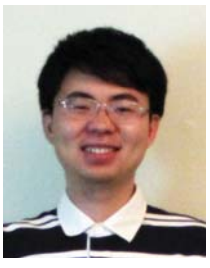
This work is partially supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office, National Basic Research Program of China (973 Program) under Grant 2012CB316400, NSF 0905084 and NSF 0917228, National Natural Science Foundation of China (Grant No: 61125203, 61173186). The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proc. of the third ACM WSDM*, New York City, 2010.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.
[3] J. Bollen, H. Mao, and A. Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proc. of the Fifth International AAAI Conference on Weblogs and Social Media*, Barcelona, Catalonia, Spain, 2011.
[4] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2011.
[5] D. Chakrabarti and K. Punera. Event summarization using tweets. In *Proc. of the Fifth International AAAI Conference on Weblogs and Social Media*, Barcelona, Catalonia, Spain, 2011.
[6] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. Technical report, 2009.
[7] T. L. Griffiths and M. Steyvers. Finding scientific topics. *The National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.
[8] D. Hall, D. Jurafsky, and C. D. Manning. Studying the history of ideas using topic models. In *Proc. the conference on Empirical Methods in Natural Language Processing*, pages 363–71, Stroudsburg, PA, 2008.
[9] G. Heinrich. Parameter estimation for text analysis. Technical report, Fraunhofer IGD and University of Leipzig, Germany, 2009.
[10] Z. Hong, X. Mei, and D. Tao. Dual-force metric learning for robust distracter-resistant tracker. In *Proc. of European Conference on Computer Vision (ECCV) 2012*, Florence, Italy, 2012.
[11] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proc. of the tenth ACM SIGKDD*, Seattle, Washington, 2004.
[12] Y. Hu, A. John, F. Wang, and D. D. Seligmann. Et-lda: Joint topic modeling for aligning events and their twitter feedback. In *Proc. of The 26th AAAI Conference on Artificial Intelligence*, Vancouver, Canada, 2012.
[13] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent twitter sentiment classification. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, 2011.
[14] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proc. of the 15th ACM SIGKDD*, Paris, France, 2009.
[15] C. X. Lin, B. Zhao, Q. Mei, and J. Han. Pet: a statistical model for popular events tracking in social communities. In *Proc. of the 16th ACM SIGKDD*, Washington, DC, 2010.
[16] F. Liu, Y. Liu, and FuliangWeng. Why is "sxsw" trending? exploring multiple text sources for twitter topic summarization. In *Proc. of the Workshop on Language in Social Media*, Portland, Oregon, 2011.
[17] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proc. the 18th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, 2002.
[18] G. Mishne and N. Glance. Predicting movie sales from blogger sentiment. In *AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs*, Stanford University, California, 2006.
[19] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proc. of the Fourth International AAAI Conference on Weblogs and Social Media*, Washington, DC, 2010.
[20] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.
[21] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proc. of the 19th international conference on World Wide Web*, Raleigh, North Carolina, 2010.
[22] D. Shahaf and C. Guestrin. Connecting the dots between news articles. In *Proc. of the 16th ACM SIGKDD*, Washington, DC, 2010.
[23] D. Tao, X. Li, X. Wu, W. Hu, and S. J. Maybank. Supervised tensor learning. *Knowledge and Information Systems*, 13(1):1–42, 2007.
[24] D. Tao, X. Li, X. Wu, and S. J. Maybank. General tensor discriminant analysis and gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1700–1715, 2007.
[25] D. Tao, X. Li, X. Wu, and S. J. Maybank. Geometric mean for subspace selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):260–274, 2009.
[26] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, 2006.
[27] Y. Tausczik and J. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2010.
[28] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment in twitter events. *Journal of the American Society for Information Science and Technology*, 62(2):406–418, 2011.
[29] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.
[30] X. Tian, D. Tao, and Y. Rui. Sparse transfer learning for interactive video search reranking. *ACM Transactions on Multimedia Computing, Communications and Applications (TOMCCAP)*, 8(3):26, 2012.
[31] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proc. of the Fourth International AAAI Conference on Weblogs and Social Media*, Washington, DC, 2010.
[32] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proc. of the 20th ACM CIKM*, Glasgow, Scotland, 2011.
[33] J. Weng and B.-S. Lee. Event detection in twitter. In *Proc. of the Fifth International AAAI Conference on Weblogs and Social Media*, Barcelona, Catalonia, Spain, 2011.
[34] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *Proc. of the fourth ACM international conference on Web search and data mining*, Hong Kong, China, 2011.
[35] W. Zhang, C. Yu, and W. Meng. Opinion retrieval from blogs. In *Proc. of the sixteenth ACM conference on Conference on information and knowledge management*, Lisboa, Portugal, 2007.
[36] L. Zhuang, F. Jing, X. Zhu, and L. Zhang. Movie review mining and summarization. In *Proc. of the 15th ACM international conference on Information and knowledge management*, Arlington, Virginia, 2006.

Shulong Tan received the BS degree in Software Engineering from Zhejiang University, China, in 2008. He is currently a PhD candidate in College of Computer Science, Zhejiang University, under the supervision of Prof. Chun Chen and Prof. Jiajun Bu. His research interests include social network mining, recommender systems and text mining.



Yang Li received the BS degrees in Computer Science from Zhejiang University, China, in 2010. He is currently a PhD candidate in the Computer Science Department of University of California at Santa Barbara, under the supervision of Prof. Xifeng Yan. His research interests include text mining, natural language understanding and data management.

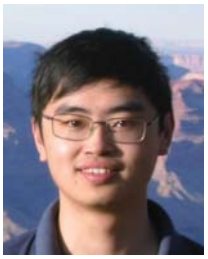
Huan Sun received the BS degree in Electronic Engineering and Information Science from the University of Science and Technology of China, in 2010. She is currently working toward the PhD degree in computer science at the University of California, Santa Barbara. Her research interests include statistical machine learning, deep learning, and data mining.

Ziyu Guan received the BS and PhD degrees in Computer Science from Zhejiang University, China, in 2004 and 2010, respectively. He had worked as a research scientist in the University of California at Santa Barbara from 2010 to 2012. He is currently a full professor in the College of Information and Technology of Chinas Northwest University. His research interests include attributed graph mining and search, machine learning, expertise modeling and retrieval, and recommender systems.

Xifeng Yan is an associate professor at the University of California at Santa Barbara. He holds the Venkatesh Narayanamurti Chair in Computer Science. He received his Ph.D. degree in Computer Science from the University of Illinois at Urbana-Champaign in 2006. He was a research staff member at the IBM T. J. Watson Research Center between 2006 and 2008. He has been working on modeling, managing, and mining graphs in bioinformatics, social networks, information networks, and computer systems. His works were extensively referenced, with over 7,000 citations per Google Scholar. He received NSF CAREER Award, IBM Invention Achievement Award, ACM-SIGMOD Dissertation Runner-Up Award, and IEEE ICDM 10-year Highest Impact Paper Award.

Jiajun Bu received the BS and Ph.D. degrees in Computer Science from Zhejiang University, China, in 1995 and 2000, respectively. He is a professor in College of Computer Science, Zhejiang University. His research interests include embedded system, data mining, information retrieval and mobile database.

Chun Chen received the BS degree in Mathematics from Xiamen University, China, in 1981, and his MS and Ph.D. degrees in Computer Science from Zhejiang University, China, in 1984 and 1990 respectively. He is a professor in College of Computer Science, Zhejiang University. His research interests include information retrieval, data mining, computer vision, computer graphics and embedded technology.



Xiaofei He received the BS degree in Computer Science from Zhejiang University, China, in 2000 and the Ph.D. degree in Computer Science from the University of Chicago, in 2005. He is a Professor in the State Key Lab of CAD&CG at Zhejiang University, China. Prior to joining Zhejiang University, he was a Research Scientist at Yahoo! Research Labs, Burbank, CA. His research interests include machine learning, information retrieval, and computer vision.