An Efficient Online Event Detection Method for Microblogs via User Modeling

No Author Given

No Institute Given

Abstract. Detecting events in microblog is important but still challenging. As tweet stream is a mixture of user interests and external events, it's expensive to distinguish them. Existing methods are ineffective since they ignore user interests or only model interests and events on a fixed dataset without scalability. In this paper, we introduce an online learning model User Modeling Based Interest and Event Topic Model (UMIETM). UMIETM (1) exploits user modeling's information to discover events by filtering out user interest-related tweets, and (2) treats the arriving data as stream and run the detection in online learning style. Furthermore, UMIETM can handle dynamic increased vocabulary in tweet stream. The UMIETM is verified on the real dataset which spans one year and contains 16 million tweets, and it outperforms a state-of-the-art model in quantitative.

1 Introduction

Detecting events in microblog is very important as the microblog has become one of the most popular sites for users to publish and get recent news. However it is still a challenging task, for the reason that tweets are (1) in large scale and (2) in a mixture of user interests and external events. Users post tweets not only for reporting breaking news, but also talking about interest related affairs. Existing methods are insufficient in this scenario, such as [1] which is ineffective when ignoring user interests, and [2] which lacks scalability when considering user interests but on a fixed dataset.

In another way, user profile or user description is vital in social media. And the existing event detection lacks the exploiting of them. Users describe themselves in profiles to show their interests, locations, occupations or identifications. These factors are stable and the high generalization of user's characteristics. One example is Biz Stone[1] whose profile is *Co-founder of Twitter, Medium, and now Co-founder and CEO of Askjelly.com.* We read his recent 100 tweets, and found 7 tweets talking about Twitter and 36 about Jelly. This example suggests that user profile is stable to reflect user's interests. As another example, the profile of a Chinese microblog user @hadoopchina[2] is *#Cloud*, *#YARN*, *#Spark*, *#Big_Data*, *#Hadoop*. He mainly tweets IT news until external events happen,

[1] https://twitter.com/biz

[2] http://weibo.com/hadoopchina

such as celebrating spring festival. This suggests profile can help to distinguish events from user interest-related tweets. Although some accounts do not have the accurate self-descriptions, they can be augmented by gathering their followings' self-descriptions.

In this paper, we introduce an online learning model, the *User Modeling Based Interest and Event Topic Model* (UMIETM) to discover the events by filtering out user interests related tweets, taking users' profiles into consideration which indicate long term interests.

It's also very interesting when we treat the event detection in microblogs as a kind of crowd sourcing service from different twitterers. As the old saying "Birds of a feather fly together", there must be something happened when all birds fly. We take user profiles as the strong signal to indicate user's behavior or interests. When different twitterers with different user profiles post or retweet something, they are telling us that we should pay attention to what they have paid attention to. We take this intuition into our event detection in microblogs.

Generally speaking, better user interests modeling means better performance of event detection, and vice versa. As the high correlation between user profiles and user interests, UMIETM can improve the effectiveness of event detection by modeling user interests with user profiles.

In summary, our contribution in this paper are two folds: (1) exploiting user profiles, which indicates users' long term interests, to discover event-related tweets, (2) proposing an online event detection method which benefits from user profiles.

2 Related Work

Study of event detection on text stream can be divided into three ways: word frequency based, text similarity based and topic model based.

Several word frequency methods have been developed for event detection such as Discrete Fourier Transform[3], wavelet analysis [4]. They treat the word's document frequency along timeline as time series and do the analysis in frequency domain. DFT method suffers the problem that it can not locate the time point for bursty. Wavelet analysis based method's complexity is very high, so its scalability is limited.

Text similarity based online event detection methods[5][6] suffer from the lexical variation which means different words describe the same events. Similarity based method can successfully detect the tweet which is retweeted by many times, but fails to find out the event which is described from many different perspectives. As a result, many events are duplicately detected due to their popularities, which may bury other events and overwhelm users with duplicated unwanted content.

In contrast, topic model can handle the lexical variation problem with word co-occurrence[7]. As many events are highly related to topics, a number of methods based on topic model have been proposed for event detection, including online detection and offline detection. Lau[1] introduces an online topic model to track emerging events in microblogs. It can deal with a massive of tweets, but it doesn't filter out the tweets related to user interests. Diao[2][8] show that event detection can benefit from filtering out user interest related tweets. And Yan[9] models the bursty topic by incorporating burstiness of biterms as prior knowledge. But they are different from ours. As these models need the whole dataset as the input, they are offline detections, which is not scalability for large dynamic dataset such as micorblogs. Instead, we gather user's self-description and the followings' self-description as user profile which is stable to characterize user interests. Based on this fine-grained user modeling information, user's long term interest related tweets and short term bursty event related tweets can be distinguished in online style, and can be efficiently applied on microblogs.

Since user modeling has the significant impact on users' activities on microblogs, the usage of user modeling has drawn attention in many research fields. [10] exploits user modeling information and network topology to infer user's role in social network. [11] also introduces user modeling to community detection where some edges are not observable, but user profile can provide additional information. Different from the above work, we do not only treat the user modeling result as an additional feature, but also an important representative factor for users, who are the source of information in microblogs.

User modeling can be carried out to obtain web users' demographics information [12] (e.g., gender, age, ethnicity, education, income, etc.). Compared with demographics information, user interests' modeling can be verified more easily [13].

3 Method

3.1 Problem Formulation

User profile. User usually describes herself or himself by a piece of short text on microblog platform. This short text can be a continuous string or an array of hash tags, e.g., "*Co-founder of Twitter, Medium, and now Co-founder and CEO of Askjelly.com*" of Biz Stone or "*#Cloud*, *#YARN*" of @hadoopchina. Though it's easy to estimate Biz Stone's interests by his self-description text, it's not always capable of doing so because some users' are very short. To overcome this limitation, we define **user profile $\boldsymbol{p_u}$** as combining user u's self-description text with the texts provided by u's followings. Taking Biz Stone as an example, he follows 696 accounts, in which there are 60 founders, 27 CEOs, 21 Google related, and 9 medium related accounts, etc. This example demonstrates that user profile p_u can be augmented by gathering the followings' information.

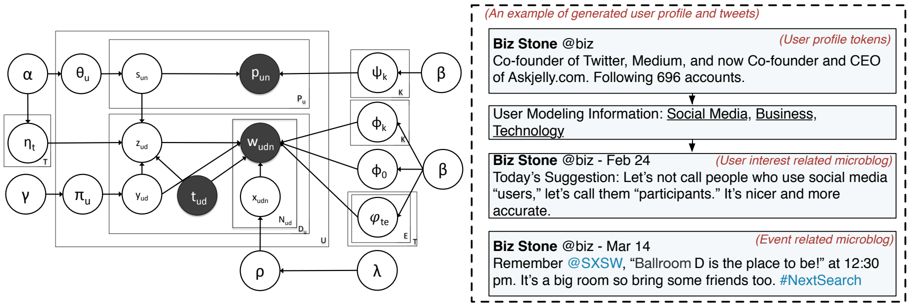
User modeling. User modeling is used to capturing user's long term interests. For each user profile token, . For example, Biz Stone's long term interests can be inferred as *Social Media*, *Business*, and *Technology* from his user profile *Medium*, *CEO*, and *Google* respectively.

The notations used in this paper are summarized in Table 1(a). We consider u's user timeline as the triple $\{uid, \boldsymbol{p_u}, \boldsymbol{w_u}\}$, where $\boldsymbol{p_u}$ represents user u's profile and $\boldsymbol{w_u} = \{(tweetid, t_{ud}, w_{ud})\}$ means the set of tweets posted by user u. The element of $\boldsymbol{w_u}$ is a triple of tweet id, time stamp t_{ud} and tweet content $\boldsymbol{w_{ud}}$.

Event. We define the event in the given time window t as the set of tweets denoted by $\{\boldsymbol{w_{te}}\}$. The event related tweets in set $\{\boldsymbol{w_{te}}\}$ hold two properties: (1) it is different from user u's long term interests and (2) it is similar with other tweets in the set. The task of event detection is to find out all the events in corpus. Different methods treat the above properties in different ways: LSH based methods[5] treat the difference and simliarity in word vector space; while the topic model based methods[2] take the semantic meaning into consideration. Under the framework of topic model, we divide the topics into user interest related topics and bursty event related topics, and further promote an online learning model UMIETM.

3.2 Model Descriptions

UMIETM is motivated by two observations: (I)user profiles are more stable to reflect users' interests than tweets; and (II)external events draw global attention in short time.



(An example of generated user profile and tweets)

Biz Stone @biz (User profile tokens)
Co-founder of Twitter, Medium, and now Co-founder and CEO of Askjelly.com. Following 696 accounts.

User Modeling Information: Social Media, Business, Technology

Biz Stone @biz - Feb 24 (User interest related microblog)
Today's Suggestion: Let's not call people who use social media "users," let's call them "participants." It's nicer and more accurate.

Biz Stone @biz - Mar 14 (Event related microblog)
Remember @SXSW, "Ballroom D is the place to be!" at 12:30 pm. It's a big room so bring some friends too. #NextSearch

Fig. 1. Illustration of UMIETM (left), and an example user Biz Stone's tweets about his long term interests and bursty event (SXSW is a music, film and interactive conference and festival hosted on March 11-20) (right).

To capture Observation I, we enhance the association between profile topic and tweet topic which is inspired by [14]. More particularly, the model generates observed data for user u in two phases as shown in Fig.1. The first phase generates the hidden topic s_{un} from user interest topic distribution θ_u, then generates profile token p_{un}.

When $y_{ud} = 0$, the second phase generates tweet topic z_{ud} from associated profile topics $\{s_{u1}, ..., s_{un}\}$ uniformly. To overcome the deficiency of user profile words, we add the smoothing factor κ for each topic to s_{un}. The second phase makes a closer correlation between profile topics and tweet topics.

For Observation II, we introduce time dependent event distribution η_t and switcher y_{ud} to distinguish user's long term interests from short term responses to external events. Only if switcher $y_{ud} = 1$, tweet topic z_{ud} will be sampled from multinomial event distribution η_t.

We introduce UMIETM's generative process and leave the detail of online learning to section 3.4. We assume that there are K latent topics corresponding to all users' interests in corpus. ψ_k and ϕ_k are profile word distribution and tweet word distribution on k-th interest topic respectively. We also assume a background word distribution ϕ_0 to filter out common words. We set E events as word distributions $\{\varphi_{te}\}$ in each time window t.

Table 1. UMIETM's Notations and Generative Process

(a) Notations

T	number of time windows
E	number of event-realted topics in each time window
K	number of interest-related topics
U	number of users
P_u	number of tokens appeared in user u's profile
D_u	number of tweets published by user u
α, β	priors of Dirichlet distributions
θ_u	K dimension vector indicating user u's interest distribution
p_{un}	user u's n-th profile token
s_{un}	the hidden topic of user u's n-th profile token
ψ_k	the user profile token's distribution on k-th profile topic
η_t	the events' distribution in time window t
π_u	the preference of user u to participate the discussion of global events
y_{ud}	the type of user u's d-th tweet ($y_{ud} = 0$ indicates interest-related, $y_{ud} = 1$ event-related)
t_{ud}	the discrete value between 1 and T indicating the timestamp of user u's d-th tweet
z_{ud}	the topic of user u's d-th tweet , $z_{ud} \in \{1, ...K\}$ if $y_{ud} = 0$; $z_{ud} \in \{1, ..., E\}$ if $y_{ud} = 1$
ϕ_k	tweet token's distribution on k-th interest-related topic
ϕ_{te}	tweet token's distribution on e-th event-related topic in time window t
x_{udn}	the boolean indicator of n-th token in user u's d-th tweet: background word if $x_{udn} = 0$; non-background word if $x_{udn} = 1$
w_{udn}	the n-th token in user u's d-th tweet: may chosen from ϕ_0, ϕ_k or ϕ_{te}
ρ	the Bernoulli parameter for boolean indicator x_{udn}
ϕ_0	tweet token's distribution on background topic
γ, λ	the priors for Bernoulli distributions

(b) Generative Process

```
$\rho \sim Bernoulli(\lambda)$, $\phi_0 \sim Dir(\beta)$
for $k$=1 to $K$:
  $\psi_k \sim Dir(\beta)$, $\phi_k \sim Dir(\beta)$
for $t$=1 to number of time windows:
  $\eta_t \sim Dir(\alpha)$
  for $e$=1 to $E$:
    $\varphi_{te} \sim Dir(\beta)$
for each user $u$:
  $\pi_u \sim Bernoulli(\gamma)$
  user interest $\theta_u \sim Dir(\alpha)$
  for $n$=1 to number of $u$'s profile tokens:
    $s_{un} \sim Multinomial(\theta_u)$
    $p_{un} \sim Multinomial(\psi_{s_{un}})$
  for tweet $d$=1 to $D_u$:
    $y_{ud} \sim Multinomial(\pi_u)$
    if $y_{ud}$=0:
      $z_{ud} \sim Uniform(\{s_{u1}, ..., s_{un}\})$
    else:
      $z_{ud} \sim Multinomial(\eta_{t_{ud}})$
    for $n$=1 to $N_{ud}$:
      $x_{udn} \sim Bernoulli(\rho)$
      if $x_{udn}$=0:
        $w_{udn} \sim Multinomial(\phi_0)$
      else:
        if $y_{ud}$=0:
          $w_{udn} \sim Multinomial(\phi_{z_{ud}})$
        else:
          $w_{udn} \sim Multinomial(\phi_{t_{ud}, z_{ud}})$
```

Overall, the generative process of user profiles and tweets in UMIETM can be described as Table 1(b).

We also propose the variant IETM to check the significance of user profile. Different from UMIETM, IETM models user interest related tweets directly. If user profile is very important to distinguish user interests from events, UMIETM will outperform IETM, vice versa.

3.3 Model Inference

We run collapsed Gibbs sampling to obtain samples of hidden variables. For space limit, we omit the detail of inference and only list the conditional distribution of each hidden variable. As there is a coupling between profile topics and tweet topics, the Gibbs sampling should be divided into two phases, first for profiles, second for tweets.

In first inference phase (shown in Algorithm 1 line 1 to line 5) we sample user profile's hidden topic s_{un} as standard LDA's collapsed Gibbs sampling[15], where $c_{uk}^{(p)}$ is the number of user u's profile words assigned to topic k, $c_{u,\cdot}^{(p)}$ is the total number of profile words of u and $c_{kv}^{(p)}$ is the times of profile word v assigned to topic k.

$$p(s_{un} = k | s_{\neg un}, \boldsymbol{p}, \alpha, \beta) \propto \frac{c_{uk}^{(p)} + \alpha}{c_{u,\cdot}^{(p)} + K\alpha} \frac{c_{kv}^{(p)} + \beta}{c_{k,\cdot}^{(p)} + V\beta} \tag{1}$$

After the convergence of first Gibbs sampling phase, we start the second (shown in Algorithm 1 line 6 to line 20). We joint sample for y_{ud} and z_{ud} using Equation(2) and (3) where $c_{kv}^{(0)}$ is the number of tweet word v assigned to k-th interest topic, and $n_{kv}^{(0)}$ is the number of word v in d-th tweet assigned to k-th interest topic. c_u^0 and c_u^1 denote the number of user u's tweets labeled as interest related and event related respectively. In order to avoid the deficiency of user profile tokens, we add smoothing parameter κ to $c_{uk}^{(p)}$ in Equation (2).

$$\begin{aligned} & p(y_{ud} = 0, z_{ud} = k | \boldsymbol{y}_{\neg ud}, \boldsymbol{z}_{\neg ud}, \boldsymbol{t}, \boldsymbol{w}, \boldsymbol{s}, \alpha, \beta, \gamma) \\ & \propto \frac{c_u^0 + \gamma}{c_u^1 + c_u^0 + 2\gamma} \frac{c_{uk}^{(p)} + \kappa}{c_{u,\cdot}^{(p)} + K\kappa} \frac{\prod_{v=1}^{V} \prod_{b=0}^{n_{kv}^{(0)} - 1} (c_{kv}^{(0)} + \beta + b)}{\prod_{b=0}^{n_{k,\cdot}^{(0)} - 1} (c_{k,\cdot}^{(0)} + V\beta + b)} \end{aligned} \tag{2}$$

$$\begin{aligned} & p(y_{ud} = 1, z_{ud} = e | \boldsymbol{y}_{\neg ud}, \boldsymbol{z}_{\neg ud}, \boldsymbol{t}, \boldsymbol{w}, \boldsymbol{s}, \alpha, \beta, \gamma) \\ & \propto \frac{c_u^1 + \gamma}{c_u^1 + c_u^0 + 2\gamma} \frac{c_{t,e}^{(1)} + \alpha}{c_{t,\cdot}^{(1)} + E\alpha} \frac{\prod_{v=1}^{V} \prod_{b=0}^{n_{tev}^{(1)} - 1} (c_{tev}^{(1)} + \beta + b)}{\prod_{b=0}^{n_{te,\cdot}^{(1)} - 1} (c_{t,e,\cdot}^{(1)} + V\beta + b)} \end{aligned} \tag{3}$$

Finally we filter out common words from semantic meaningful words by Equation (4) and (5). Here the hidden variable $x_{udn} = 0$ indicates that w_{udn} is a common word. $c_v^{(B)}$ is the times of tweet word v assigned to the background topic. M_0^ρ is the total number of common words in corpus and $M_0^\rho + M_1^\rho$ equals the total number of tokens.

$$\begin{aligned} & p(x_{udn} = 0 | x_{\neg und}, w_{udn} = v, w_{\neg udn}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{t}, \alpha, \beta, \gamma, \lambda) \\ & \propto \frac{M_0^\rho + \lambda}{M_0^\rho + M_1^\rho + 2\lambda} \frac{c_v^{(B)} + \beta}{\sum_{v=1}^{V} c_v^{(B)} + V\beta} \end{aligned} \tag{4}$$

$$\begin{aligned} & p(x_{udn} = 1 | x_{\neg und}, w_{udn} = v, w_{\neg udn}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{t}, \alpha, \beta, \gamma, \lambda) \\ & \propto \frac{M_1^\rho + \lambda}{M_0^\rho + M_1^\rho + 2\lambda} (\frac{c_{k,v}^{(0)} + \beta}{c_{k,\cdot}^{(0)} + V\beta})^{I(y_{ud}=0)} (\frac{c_{t,e,v}^{(1)} + \beta}{c_{t,e,\cdot}^{(1)} + V\beta})^{I(y_{ud}=1)} \end{aligned} \tag{5}$$

Algorithm 1 UMIETM batch learning algorithm

```
 1: initiate the topic label and the statistics
 2: for $i = 1 : I_1$ do
 3:     for $u$ in user set U do
 4:         for $n = 1 : P_u$ do
 5:             sample profile's hidden topic $s_{un}$ by (1)
 6:             update $s_{un}$, $c^{(p)}_{u,k}$ and $c^{(p)}_{k,v}$
 7: for iteration $i = 1 : I_2$ do
 8:     for $t = 1 : T$ do
 9:         for $u$ in user set $U_t$ do
10:             for $d = 1 : D_u$ do
11:                 sample $y_{ud}$ and $z_{ud}$ by (2), (3)
12:                 if $y_{ud} = 0$ then
13:                     update $z_{ud}$, $y_{ud}$, $c^{(0)}_u$, $c^{(0)}_{u,k}$, $c^{(0)}_{k,v}$
14:                 else
15:                     update $z_{ud}$, $y_{ud}$, $c^{(1)}_u$, $c^{(1)}_{t,k}$, $c^{(1)}_{t,k,v}$
16:                 for $n$ in $1, \cdots, N_{ud}$ do
17:                     sample $x_{udn}$ by (4), (5)
18:                     if $x_{udn} = 0$ then
19:                         update $x_{udn}$, $M^{\rho}_0$, $c^{(B)}_v$
20:                     else
21:                         update $x_{udn}$, $M^{\rho}_1$, $c^{(0)}_{k,v}$, $c^{(1)}_{t,k,v}$
```

Algorithm 2 UMIETM online learning algorithm

```
 1: for all $u \in \mathcal{U}$, load $\boldsymbol{p_u}$ and $\boldsymbol{w_u}$
 2: for all $u \in \mathcal{U}$, $k$, $v$, load $M^{\rho}_0$, $M^{\rho}_1$,$c^{(p)}_{u,k}$, $c^{(p)}_{k,v}$,
    $c^{(0)}_u$, $c^{(1)}_u$, $c^{(0)}_{u,k}$, $c^{(0)}_{k,v}$, $c^{(B)}_v$ from trained Model $\mathcal{M}$.
 3: for $t = 1 : T$ do
 4:     update the vocabulary for profile and tweet
 5:     for iteration $i = 1 : I_1$ do
 6:         for $u$ in user set $\mathcal{U} \cup U_t$ do
 7:             do operation as line 4 to line 5 in Algorithm 1
 8:     $\mathcal{U} = \mathcal{U} \cup U_t$
 9:     for iteration $i = 1 : I_2$ do
10:         for $u$ in user set $U_t$ do
11:             do operation as line 9 to line 20 in Algorithm 1
```

3.4 Online Learning on Tweet Stream

Gibbs Sampling on fixed large dataset is very expensive both in memory and time. Each Gibbs sweep need to maintain 12 statistics such as $c^{(p)}_{k,v}$, $c^{(p)}_{u,k}$ appeared in Equation(1) to (5). Generally, the complexity of the Gibbs sampling is $O(IK|W|)$ where I_1, I_2 are iteration numbers for profiles and tweet tokens, K is topic number, E is event number, $|P|$ is number of profile tokens and $|W|$ is number of tweet tokens. More important, we have to maintain all tweets in memory for batch learning but it is unacceptable.

We propose the online learning method shown in Algorithm 2, and denote the previous learned model as $\mathcal{M}$, previous trained users $\mathcal{U}$. We update $\mathcal{M}$ increasingly by tweets in each time window.

There are two tricks in our online method. The first one is line 6 and line 8 in Algorithm 2 which runs the batch sampling for all users' profile. User profile can reflect user's interests better than tweets and the user profile token number $|P|$ is much smaller than $|W|$. The second one is line 10 which run the sampling only on the tweets in current time window. It can filter out interest-related tweets by user profile, and detect event-related tweets.

Finally, we handle the dynamic increased vocabulary in line 4. Take $\phi_{k,v}$ as an example, when we meet the word v unseen, $\phi_{k,v}$ in current time window can be initially estimated as $\beta/(c^{(0)}_{k,\cdot} + V'\beta)$, and other words appeared in previous time windows can be estimated as $(c^{(0)}_{k,v} + \beta)/(c^{(0)}_{k,\cdot} + V'\beta)$. And V' is the size of increased vocabulary.

4 Experiment

Here we present the effectiveness of our proposed algorithm UMIETM and the efficiency of its online performance. We evaluate the efficiency by perplexity, precision for event detection. We check the time cost and complete likelihood for efficiency.

Weibo is a popular Chinese microblogging service[3]. We crawl weibo data by its public API[4] from Jan 2012 to Dec 2012. To improve the quality of analyzing on tweets, we do necessary pre-processing: (1) splitting dataset by week, (2) segmenting Chinese words, (3) removing stop words and low frequency words whose document frequency in its time window is less than 3, (4) removing tweets whose token number is less than 3. To model user interests better, we remove users from dataset who has less than 2 hashtags in profile. After pre-processing we get 252 thousand users, 16 million tweets and 251 million tweet tokens listed in Table 2(a).

Table 2. statistics of processed dataset

(a) Weibo Dataset

	#user	#tweet
whole year	252,369	16,421,167
week1	9,785	31,503
week2	29,721	242,554
week3	30,891	254,698
week4	29,788	237,456

(b) Metrics of event detection

	precision	recall
UMIETM	0.894	0.913
UMIETM(-)	0.847	0.697
IETM	0.824	0.536
LSH	0.394	0.913
EDCoW	0.731	0.435

(c) Held out perplexity

Author-LDA	twitterLDA	timeUserLDA	IETM	UMIETM
20422.25	6027.47	4810.92	3926.76	3107.83

We compare our model UMIETM with twitterLDA[16], timeUserLDA[2], Author-LDA, and our model's variant IETM. Author-LDA combines the tweets posted by same author into a single document, then run standard LDA on the assembled tweets. TwitterLDA is designed for topic modeling on twitter. We compare with Author-LDA and TwitterLDA for confirming the significance of distinguishing user interests from events in microblog. TimeUserLDA[2] is designed for retrospective event detection in microblog, and considers to distinguish events from user interests. We compare with timeUserLDA to show the impact of user profile. The variant IETM (*Interest and Event Topic Model*) models user's interests and events without the help of user profiles.

We set the asymmetric α and symmetric $\beta = 0.01$ for UMIETM, where α will be optimized by Gibbs EM algorithm[17]. $\alpha = 0.1$, $\beta = 0.01$ are set for

[3] http://en.wikipedia.org/wiki/Sina_Weibo/

[4] http://open.weibo.com/

all remaining models. After cross validation we find that UMIETM and IETM perform best on $K = 90$ and $E = 30$, $\kappa = 0.01$. To compare equally, we set the same topic number for Author-LDA, twitterLDA, timeUserLDA (which means when we run models offline on 1 time window, we shall set 120 topics for all, and 150 topics on 2 time windows and so on).

To verify the role of user profiles played, we set UMIETM(-) as the degradation of UMIETM, which take symmetric prior α.

4.1 Effectiveness

In this subsection, we illustrate the performance of UMIETM in which user profile is considered.

Quantitative Measure. We initialize UMIETM and IETM by batch learning on data from first week to third week, then run them in online learning way from fourth week to ninth week. On each week we calculate their perplexities[18], where $perplexity(D_{test}) = \exp\{-\frac{\sum_{u=1}^{U}\sum_{d=1}^{D_u}\log p(w_{ud})}{\sum_{u=1}^{U}\sum_{d=1}^{D_u} N_{ud}}\}$ and $p(w_{ud}) = (1-\pi_u)\sum_{k=1}^{K}\theta_{uk}\prod_{n=1}^{N_{ud}}(\phi_{s,w_{udn}}(1-\rho)+\phi_{k,w_{udn}}\rho)+\pi_u\sum_{k=1}^{K}\eta_{t,k}\prod_{n=1}^{N_{ud}}(\phi_{s,w_{udn}}(1-\rho)+\phi_{k,w_{udn}}\rho)$. The others are trained from first week to third week, and the held out perplexities are calculated on data from fourth week to ninth week. TimeUserLDA and IETM's perplexities are smaller than User-LDA, twitterLDA as they both consider distinguishing user interests from events.

In Table 2(c) the perplexity of UMIETM is 3107.8, and much smaller than others. It demonstrates that user profile is significant for tweet stream's modeling.

In Table 2(b), we evaluate the events detected by models. We asked the annotators to label the event with score 1, and non-event related topic as 0. The precision and recall is illustrated, where UMIETM(-) performs slightly better than IETM. And UMIETM outperforms UMIETM(-) in detecting high quality event. A reasonable analysis is that UMIETM uses the profile information sufficiently by asymmetric priors[19]. In this way, we also prove that the well exploiting of user profile information is important to model user interests and events on tweets.

Case Study Some events detected by our UMIETM model are shown in Table 3. Comparing with UMIETM, timeUserLDA fails to discover the *shoddy construction* event in the second week, while IETM reports this event as *bi, women, elegant, adoption, engineering, reed.* Obviously IETM fails to distinguish this event from user interests. UMIETM filters users' interests like *bi, women, elegant, adoption* using their profile #baby, #women, and detect the event.

4.2 Efficiency

We implement our methods on mallet[5], and run them on Linux server with 8 cores(2.00GHz) and 64GB memory. In this subsection, we verify the convergence and online performance of our online learning algorithm.

[5] http://mallet.cs.umass.edu/dist/mallet-2.0.7.tar.gz

Table 3. Example Events detected by UMIETM

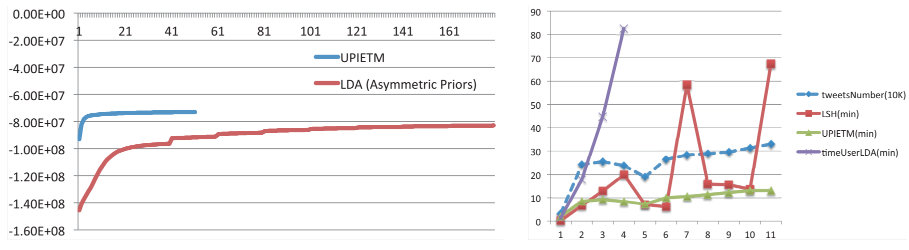
Time window	Top words for Event Example	Event
The first week of 2012	Japan, earthquake, occur, the first day, January, 7.0, 2012	In January 1 of 2012, a magnitude-7 earthquake occurred in Japan.
	New, year, happy, 2012, New Year's Day, healthy, blessing, happiness	Everyone blesses happy new year in the first day of 2012
The second week of 2012	reed, steel, Engineering, appearance, engineering, shoddy construction, criminal	In an accident, a car crashed through the guardrail into the river. People found that,the guardrail was built with reed which should be built with steel bar.
The third week of 2012	Apple, 4S, iPhone, line up, scalper, Sanlitun	Many scalpers lined up to buy the Apple iPhone 4S when it started to sell at Sanlitun Apple store in January 13th.

Convergence. The convergence of algorithm is vital in real time streaming environment. UMIETM needs several Gibbs sweeps to find out the optimal parameters of model. But the size of data is usually large even spitted into time windows. The time complexity of UMIETM is $O(I_1K|P| + I_2K|W| + I_2E|W|)$ given in section 3.4. It suggests that the more iterations Gibbs sampling have to do, the less efficiently our model performs. Fortunately UMIETM is very economic to its stable state and does not need so much iterations. As suggested by [19] and [20], we choose LDA with asymmetric priors as baseline, which is much more comparable than standard LDA on microblog dataset. We run UMIETM on the first three time windows. Correspondingly we only use the tweets in these time windows for LDA with asymmetric priors. Burn-in period is set to 20 and optimize interval also 20, which means the priors are optimized every 20 iterations as the red curve in Fig 2(a).

One of the criteria for stopping Gibbs sampling is that the complete log likelihood is stabilized. We stop UMIETM after 50 iterations since the complete log likelihood only increases 0.02% in last iteration. It will take LDA(asymmetric priors) 300 minutes to run 1000 iterations, and 20 minutes per 10 iterations for UMIETM. There is a trade-off between the effectiveness by more iterations and efficiency by less, we set the iteration number as 10 in online setting. After 10 iterations the UMIETM's complete log likelihood will increase no more than 0.2%.

Online Performance. As verified in the above subsection *Convergence*, we run 10 Gibbs sweeps for each time window in online learning phase. When UMIETM is implemented in online way, it doesn't have to revisit the tweets in previous windows. So its time cost in each window is proportional to the data size of current window as shown in Figure 2(b). In average, online UMIETM can process five thousand tokens per second.

LSH based event detection on microblogs is used by [5],[6]. The former one detects events as the cluster of tweets by LSH, and the latter one extends this method on distributed machines. They both only use the cosine similarity without considering semantic meaning of words. So LSH based method may split the same event into clusters of similar tweets. However LSH based method reports 10578 events in 7th time window, and 10901 events in 8th window. The other problem of LSH based event detection method is that the bad design may lead unstable performance. The tweets are mapped into LSH defined space in a skew way which is illustrated in Fig 2(b). The other comparison is timeUserLDA[2] which is designed for retrospective event detction. The purple line in Fig 2(b) goes up straightly as timeUserLDA has to revisit previous tweets to make a decision whether the tweet is event related. The perplexity of online UMIETM on the 11 time window in Fig 2(b) is 3036.44 ± 397.14. And the average precision of detecting high quality event is 0.22. Both indicators demonstrate that UMIETM can perform well in online learning way.



(a) $\alpha = (2.5, 2.5, 3.5)$

(b) $\alpha = (0.25, 0.75, 0.75)$

Fig. 2. (a) The convergence of complete log likelihood of UMIETM and LDA(with asymmetric priors). The x-axis represents the round of iteration and y-axis shows the log likelihood. (b) Efficiency of UMIETM. The x-axis represents the time window, and y-axis shows the duration of processing or the number of tweets in corresponding time window.

5 Conclusions

Microbiolog is mixed with user interests and external events. The quality of event detection in microblog depends on how we distinguish them. We exploit user profile to discover events by filtering out user interest-related tweets. We further treat the arriving data as stream and run the detection in online learning style. The experiments demonstrates that our method is effective and efficient for online event detection in microblogs.

References

1. Jey Han Lau, Nigel Collier, and Timothy Baldwin. On-line trend analysis with topic models:#twitter trends detection topic model online. In *COLING*, 2012.
2. Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. Finding bursty topics from microblogs. In *ACL*, 2012.

3. Qi He, Kuiyu Chang, and Ee-Peng Lim. Analyzing feature trajectories for event detection. In *ACM SIGIR conference on Research and development in information retrieval*, pages 207–214. ACM, 2007.
4. Jianshu Weng and Bu-Sung Lee. Event detection in twitter. In *ICWSM*, 2011.
5. Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *HLT-NAACL*, 2010.
6. Richard McCreadie, Craig Macdonald, Iadh Ounis, Miles Osborne, and Sasa Petrovic. Scalable distributed event detection for twitter. 2013.
7. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *JMLR*, 2003.
8. Qiming Diao and Jing Jiang. A unified model for topics, events and users on twitter. In *EMNLP*, 2013.
9. Xiaohui Yan, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. A Probabilistic Model for Bursty Topic Discovery in Microblogs. *AAAI*, pages 353–359, 2015.
10. Yuchen Zhao, Guan Wang, Philip S Yu, Shaobo Liu, and Simon Zhang. Inferring social roles and statuses in social networks. In *SIGKDD*. ACM, 2013.
11. Tetsuya Yoshida. Toward finding hidden communities based on user profile. *Journal of Intelligent Information Systems*, 40(2):189–209, 2013.
12. Aron Culotta, Nirmal Ravi Kumar, and Jennifer Cutler. Predicting the demographics of twitter users from website traffic data. In *AAAI*, pages 72–78, 2015.
13. Stefano Faralli, Giovanni Stilo, and Paola Velardi. Large scale homophily analysis in twitter using a twixonomy. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2334–2340. AAAI Press, 2015.
14. David M Blei and Michael I Jordan. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 2003.
15. Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *PNAS*, 2004.
16. Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*. 2011.
17. Hanna M Wallach. Structured topic models for language. *Unpublished doctoral dissertation, Univ. of Cambridge*, 2008.
18. Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *ICML*, 2009.
19. Hanna M Wallach, David M Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *NIPS*, volume 22, pages 1973–1981, 2009.
20. Ming Zhang Xuanlong Nguyen Qiaozhu Mei Liebig's Barrel Jian Tang, Zhaoshi Meng. Understanding the limiting factors of topic modeling. In *ICML*, 2014.