2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing

Online Bursty Event Detection from Microblog

Jianxin Li, Zhenying Tai, Richong Zhang, Weiren Yu
State Key Laboratory of Software Development Environment
School of Computer Science & Engineering
Beihang University, Beijing, China
{lijx,taizy,zhangrc,yuweiren}@act.buaa.edu.cn

Lu Liu
School of Computing and Mathematics
University of Derby
Derby, UK
l.liu@derby.ac.uk

***Abstract*—Microblogs (e.g., Twitter and Weibo) have become a large social media platform for users to share contents, their interests and events with friends. A surge of the number of event related posts always reflects that some people's concern real-life events happened. In this paper, we propose an incremental temporal topic model for microblogs namely** BEE **(Bursty Event dEtection) to detect these bursty events.** BEE **supports to detect these bursty events from short text datasets through modeling the temporal information of events. And** BEE **employs processing the post streaming incrementally to track the topic of events drifting over time. Therefore, the latent semantic indices are preserved from one time period to the next. After** BEE **detects the event-driven posts and related events, the bursty detection module can identify the bursty patterns for each event and rank the events using the bursty patterns. Our experiments on a large Weibo dataset show that our algorithm can outperform the baselines for detecting the meaningful bursty events. Subsequently, we also show some case studies that indicate the effectiveness of the temporal factor for bursty event detection and how well** BEE **can track the topic drifting of events.**

Keywords*—*event detection; topic drifting; online;

I. Introduction

Bursty event detection from microblogs has received considerable attentions in the fields of data mining and knowledge discovery, because of the information that posts in microblogs are real-time and often event-driven[5]. Specifically, the textual content coupled with the temporal patterns of these microblog posts provide an important insight into the general interest of the public. A sudden increase of topically similar posts usually indicates a burst of interests in some events that have happened. Online bursty event detection from microblogs therefore can help us identify the popular events that have been drawing the public's attention largely.There is thus an urgent need to provide users with tools which can automatically extract and summarize significant information from highly dynamic social streams [9]. However, unlike the traditional normal documents (e.g. news articles, academic papers), the lack of rich context in short texts [20] and the topic of event drifting over time make online event detection a challenging problem.

Existing studies on detecting bursty events are based on detecting bursty keywords. In general, the bursty events are represented by a group of related bursty keywords (or segmentations) with the bursty temporal informations. Some clustering methods (e.g., Graph partitioning, Density-based K-means) are used to form events. The relation of each two keywords can be calculated using the co-occurrences of the words. While the cosine distance of vectors are used to measure the similarity of each two posts. For example, Michael and Nick [13] detect events by grouping the bursty keywords. However, the methods above are not built upon robust probabilistic background. And they have ignored the topical similarity between keywords. In addition, they can not model the topic drifting of events.

Recently, another way to deal with the problem are based on the topic models, like PLSA[7] and LDA[1], which are widely used for uncovering the hidden variables from the text collections. The occurrence of words can be modeled with the probabilistic theory in these methods and the topical similarity between keywords can be measured. Conventional topic models reveal the latent variables within the text corpus by implicitly capturing the document-level word co-occurrence patterns [2][16]. Therefore, directly applying these models on short texts will suffer from the severe data sparsity problem (i.e. the sparse word co-occurrence patterns in each short document) [8]. The limited contents make it difficult for topic models to identify the topics in short documents. Therefore, the conventional topic models and some variants of them can not deal with our problem. Moreover, the drifting of event topic over time can not be measured and the temporal information does not be modeled.

In this paper, we propose an incremental temporal topic model (BEE) which is designed for detecting bursty events from microblogs. BEE can model the temporal information of events in microblogs and identify the bursty events based on the temporal information. To adapt to the scenario of online bursty event detection, BEE employs processing the post streaming incrementally, and BEE can track the topic of events drifting over time. And the latent semantic indices are preserved from one time period to the next. The experiments show that BEE can detect more meaningful events than the baselines and depict topic drifting well.

The remainder of this paper is structured as follows: In Section 2, we introduce some related works. In Section 3, we describe our model (BEE). Section 4 details the experiments results. Then, in Section 5, we present our conclusions.

II. Related Work

Recently, event detection on microblogs stream becomes a hot research topic. Swit and Tsuyoshi [14] focused on detecting the breaking news from twitter. The method of event detection and tracking in their approach are employed by clustering the similar tweets together. The similarity of two tweets of breaking news are measured by using a variant of the TF-IDF (term frequency - inverse document frequency). But their algorithm only deal with the tweets with hashtag #breakingnews. Li et al. [11] proposed an event detection and analysis

978-1-4799-7881-6/14 $31.00 © 2014 IEEE
DOI

system to detect crime and disaster related events (CDE) from tweets. Their system supports to extract a geographic location for events and rank events based on the importance of events. Sakaki et al. [15] idenfitied the bursty events about earthquake happened in Japan. Support Vector Machine (SVM) was used to judge whether each tweet to be about earthquake or not. To summarize, the above approaches are applicable to certain types of tweets (e.g., having a specific hashtag, related to crime and disaster or containing the certain keywords). Some priori knowledge about the events to be detected is needed. So, they can not detect the general events for our problem without any priori knowledge.

Several other people make a few of efforts to detect the unspecific events without priori knowledge. Michael and Nick [13] group bursty keywords into events according to their co-occurrences in history tweets. Wei et al. [19] identify bursty co-occurrences between keywords quickly using dimension reduction, and detect the bursty events based on the co-occurrences by solving the optimizaion problem. Li et al. [10] split each tweet into non-overlapping segments (semantically meaningful information units). They clustered the bursty segments as event segments. However, the above systems ignore the temporal information of events and can not track the changes of events. So, the bursty importance of events and changes of events are not measured.

Some other event detection approaches are based on the temporal information of the events. Weng and Lee [18] apply wavelet transformation to fit the temporal information of each word. The events are formed by using modularity-based graph partitioning algorithm. However, the similarity between words measured by the cross correlation between signals can not differentiate the busty distinct events happened at the same period by coincidence. Diao et al [5] proposed a LDA-based topic model that exploits this idea to find the bursty global events. However, their model is not immediately applicable for our problem. Firstly, it is infeasible to model the interest of all the users, because most of the users release very little tweets. Secondly, their model detect the bursty events offline, it is not suitable for our online analysis scenario.

To solve the above problems, we proposed BEE. BEE can distinguish the events that happened around the same period well by modeling the temporal information of events with the observation that the topical similar contents appear around the same time. Furthermore, BEE can also depict the topic drifting of event and track the events evolution over time well.

III. BEE

A. Preliminaries

1, Notation Description We first introduce the notations used in this paper and formally formulate our problem. We assume that we have a stream of D microblogs posts, denoted as $d_1, d_2, ..., d_D$. Each post d_i is also associated with a discrete timestamp $t_d = t$, where t is an index between 1 and T. T is the total number of time points we consider. Each document d_i is denoted as a bag of words $\{w_{i,1}, w_{i,2}...w_{i,j}...w_{i,N_i}\}$, where $w_{i,j}$ is an index between 1 and V, and V is the size of the vocabulary. N_i is the number of the words in document d_i.

2, Problem Formulation A bursty event is a topic where 1, there is a surge in the number of posts related to the event. 2, it is popular in the posts during the period when it surges. Given the post stream D, our task in this paper is to detect the bursty events from D by using an online algorithm and implementation.

3, Probabilistic Latent Semantic Indexing Algorithm PLSA is a statistical view on LSA (Latent semantic analysis). In contrast to standard LSA, its probabilistic variant has a sound statistical foundation and defines a proper generative model of the data. [7] In a sense, it does capture the possibility that a document may contain multiple topics since $p(z|d)$ servers as the mixture weight of the topics for a particular document d. [1]

B. The Generative Process

What the different between PLSA and BEE is that BEE follows the two assumptions: 1, one post is always related to one event. 2, the temporal information of the topically similar posts always appear around the same time. In BEE, we assume that there are K latent topics in the text stream, d denotes a document, w denotes a term in a document, z denotes a latent variable, t_d denotes the released time of document d. In order to explain all the words in the streaming, including the meaningless words and the common words. We further assume a background word distribution θ_B over common words. All the words in each post are generated from some mixture of these $K + 1$ underlying topics.

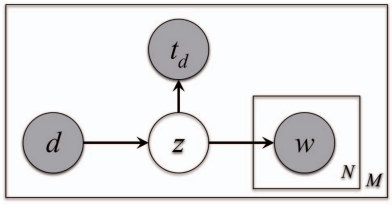

Fig. 1: Graphical model representation of BEE

In standard topic model, such as PLSA and LDA, a document is generated by a mixture of topics, represented by a distribution over topics. This is a reasonable assumption for long documents. But in microblogs, the posts often contains one or several sentences. Consequently, a single post is most likely to be about one event or meaningless. We therefore associate a single hidden variable and the background topic with each post to indicate its topic. The assumption of assigning a single topic to a short sequence of words has been used before [5][6][22].

As we have discussed in Section I, the topically similar contents are more likely published around the same time.To model this observation, we assume that there is a distribution over time points for each topic. The higher the probability for a topic at one time point is, the more popular the topic is at that time.

In general, the Fig. 1 shows the graphical model representation of BEE, the joint distribution of the word vector $\vec{w}$ and the time t_d in document d would be regarded as a sample

drawn from the following mixture model:

$$P(\vec{w}, t_d|d) = \lambda_{\theta_B} p(t_d|\theta_B) p(\vec{w}|\theta_B) + (1 - \lambda_{\theta_B}) A$$
$$A = \sum_z \{p(z|d) p(t_d|z) \prod_w p(w|z)^{n(w,d)}\} \quad (1)$$

where λ_{θ_B} is the mixture weight of the background model, and the $p(t_d|z)$ is the intensity of topic z at the time point t.

The log-likehood of the generating text sample is thus:

$$log\mathcal{L} = \sum_d p(d)\{\lambda_{\theta_B} p(t_d|\theta_B) p(\vec{w}|\theta_B) + (1 - \lambda_{\theta_B}) A\}$$
$$A = \sum_z \{p(z|d) p(t_d|z) \prod_w p(w|z)^{n(w,d)}\} \quad (2)$$

Parameter Estimation : The standard procedure for maximum likelihood estimation in latent variable models is the Expectation Maximization(EM) algorithm[7]. The expectation (E) step where posterior probabilities are computed for the latent variables [7]. In our model, according to the Bayes rule, the conditional probability of $p(z|\vec{w}, d, t_d)$ can be estimated by the following in the **E (Estimation)** step:

$$p(z|\vec{w}, d, t_d) = \frac{p(z|d) p(t_d|z) \prod_{w \in d} p(w|z)^{n(w,d)}}{\lambda_{\theta_B} p(t_d|\theta_B) p(\vec{w}|\theta_B) + B}$$
$$B = (1 - \lambda_{\theta_B}) \sum_z \{p(z|d) p(t_d|z) \prod_{w \in d} p(w|z)^{n(w,d)}\} \quad (3)$$

where the $\vec{w}$ denotes as the words vector in the document d. $n(w, d)$ denotes the frequency of the word w in the document d. The $p(w|z)$, $p(z|d)$, $p(t_d|z)$ have been estimated in the previous M-step. And the λ_{θ_B}, $p(t_d|\theta_B)$ and $p(\vec{w}|\theta_B)$ have been initialized before the iteration start.

In the **M (Maximization)** step, the probabilities $p(w|z)$, $p(t_d|z)$ and $p(z|d)$ can be estimated, respectively, by the following:

$$p(w|z) = \frac{\sum_d n(w, d) p(z|\vec{w}, d, t_d)}{\sum_w \sum_d n(w, d) p(z|\vec{w}, d, t_d)} \quad (4)$$

$$p(z|d) = \frac{p(z|\vec{w}, d, t_d)}{\sum_z p(z|\vec{w}, d, t_d)} \quad (5)$$

$$p(t_d|z) = \frac{\sum_d p(z|\vec{w}, d, t_d)}{\sum_t \sum_d p(z|\vec{w}, d, t_d)} \quad (6)$$

where the $p(z|\vec{w}, d, t_d)$ have been estimated in the previous E-step. $n(w, d)$ denotes the frequency of the word w in the document d.

C. Incrementally Update Parameters

After the completion of the above training process, estimated $p(w|z)$ parameters are used to estimate new parameters $p(z|q), p(t_q|z), p(z|\vec{w}, q, t_q)$ for a new document q in the data streaming. In this process, new parameters can be calculated using equation (4)(5)(6).Note that in the EM procedure, the $p(w|z)$ remain fixed for online event detection.

In the scene of online bursty event detection, the topic of event drifts over time. A new document may actually indicate a turning point in the story line of the event. And intuitively, the old inactive event less likely attracts new documents than a recently active event. Consequently, the parameters learned from the history data are not suitable for all the data in the streaming. To follow this changes, incorporating a lookup window is a popular way of limiting the time frame that an incoming document can relate to. Our model revises the

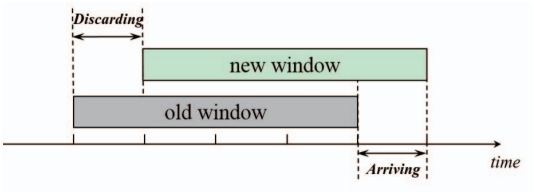

Fig. 2: The discarding and arriving process

parameters based on the posts in a sliding-window as shown in Fig.2 via the following three steps periodically using the similar ideas by Chou [4].

1) Discard old documents: As the time window slides forward, our algorithm removes the out-of-date posts d_{old} and removes the terms w_{old} that are not used in recent posts. So, the parameters $p(w_{old}|z)$, $p(z|d_{old})$ and $p(t_{d_{old}}|z)$ are also removed. Subsequently, the remaining parameters must be renormalized proportionally as follows:

$$p(t_d|z) = \frac{p(t_d|z)}{\sum_t p(t_d|z)} \quad (7)$$

$$p(w|z) = \frac{p(w|z)}{\sum_w p(w|z)} \quad (8)$$

$$p(z|d) = \frac{p(z|d)}{\sum_z p(z|d)} \quad (9)$$

2) Initiate the parameters: In this step, new documents and terms are arriving from the data streaming. We estimate $p(w_{new}|z)$ and $p(z|\vec{w}, d_{new}, t_d)$ in the following steps. To estimate $p(w_{new}|z)$, the $p(z|\vec{w}, d_{new}, t_d)$ must be estimated in the first. Firstly, the $p(w|z)$ are fixed to be used for estimating $p(z|\vec{w}, d_{new}, t_d)$ in (10). The w_{old} represent the words appear in both d_{new} and the documents in the time window simultaneously.

$$p(z|\vec{w}, d_{new}, t_d) = \frac{p(z|d_{new}) p(t_{d_{new}}|z) C}{\sum_z \{p(z|d_{new}) p(t_d|z) C\}}$$
$$C = \prod_{w_{old} \in d_{new}} p(w|z)^{n(w, d_{new})} \quad (10)$$

$$p(z|d_{new}) = \frac{p(z|\vec{w}, d_{new}, t_d)}{\sum_z p(z|\vec{w}, d_{new}, t_d)} \quad (11)$$

$$p(t_{d_{new}}|z) = \frac{\sum_{d_{new}} p(z|\vec{w}, d_{new}, t_d)}{\sum_t \sum_{d_{new}} p(z|\vec{w}, d_{new}, t_d)} \quad (12)$$

Secondly, all the $p(z|d_{new})$ and $p(t_{d_{new}}|z)$ have been calculated, $p(w_{new}|z)$ are initialized randomly and normalized. Meanwhile, the $p(z|d_{new})$ and $p(t_{d_{new}}|z)$ are fixed and used

to estimated $p(w_{new}|z)$ for new terms. The EM algorithm is applied in the process.

$$p(z|\vec{w}, d, t_d) = \frac{p(z|d)p(t_d|z)\prod_{w\in d} p(w|z)^{n(w,d)}}{\sum_z\{p(z|d)p(t_d|z)\prod_{w\in d} p(w|z)^{n(w,d)}\}} \quad (13)$$

$$p(w_{new}|z) = \frac{\sum_d n(w,d)p(z|\vec{w}, d, t_d)}{\sum_w \sum_d n(w,d)p(z|\vec{w}, d, t_d)} \quad (14)$$

3) Parameters Revising: After all the parameters are initiated in the previous step, We adjust the parameters $p(w|z)$ by discarding the out-of-date terms and appending new terms. All $P(w|z)$ are normalized using the following (7),(8),(9). Then, the EM algorithm is used, as described in (3),(4),(5),(6) to revise all the BEE parameters, because new terms w_{new} and new documents d_{new} have been introduced.

The advantages of the incremental updating parameters are twofold. First, the EM algorithm for our model will convergence more rapidly than reestimating all the data, because most of the parameters are modified slightly. The second advantage is that the continuity of the latent semantic variables is maintained. This is important for tracking the event. Based on the advantages, our algorithm is more effective and efficient than the non-incremental algorithm.

D. Bursty Detection

Just like other topic models, such as PLSA and LDA, Our model dose not generate the bursty events directly. What our model generate is a set of events E^c and the probability of each event evolves over time denoted as $\{p(t_1|e), p(t_2|e)...p(t_n|e)\}$. In our model, $p(t_i|e)$ denotes the probability of the event e at the time t_i, which represents the intensity of the event z at time point t_i. The greater $p(t_i|e)$ is, the more tweets are released about the events e, and the greater public's interest is aroused. To measure the bursty important events from the temporal information of events, we use the following mechanism, which is based on the idea by TwitInfo [12]. we wish to identify each time point i such that $p(t_i|e)$ is large relative to the recent history $p(t_{i-1}|e), p(t_{i-2}|e)...$, not the local maxima amongst the sequence of all the $p(t|e)$. The analogue in our framework is that the algorithm must determine whether a time point has an unusually large number of tweets in it. We take inspiration from TCP' s congestion control mechanism which must determine whether a packet is taking unusually long to be acknowledged and is thus an outlier. To summarize the algorithm: when the algorithm encounters a significant increase in bin count relative to the historical mean, it starts a new window and follows the increase to its maximum. The algorithm ends the peak' s window once the bin count returns to the same level it started at, or when it encounters another significant increase[12]. We can use online algorithm to implement the processing. After the bursty detection, we rank the events containing the bursty information in the latest time points based on the probability of the event e. For our model, $p(e)$ represent the number of posts related to the topic e are there in the dataset.

IV. Experiments

In the section, we show the effectiveness evaluation of our algorithm. Firstly, we show that our proposed method can identify more meaningful bursty event patterns to reveal the major real-world events from the data streaming than the PLSA (Probabilistic Latent Semantic Analysis) and TM (TwitterMonitor) baselines. Secondly, we illustrate the advantages of our model by sampling some results as the case studies analysis. Lastly, the topic drifting result shows that BEE can depict the topic drifting well.

1) TM: (TwitterMonitor)[13]: It is a system that performs trending topic detection over the Twitter stream. We implemented two core algorithms of this paper (bursty keywords detection and grouping the keywords). In this experiment, we set the size of the window to one day for both these algorithm and our model.

2) PLSA: (Probabilistic Latent Semantic Analysis)[7]: It is a classic latent semantic analysis algorithm, our model is based on the algorithm. In this model, What the different between PLSA (Probabilistic Latent Semantic Analysis) and BEE what the different between it and BEE is that it drops the two assumptions: 1, one post is always related to one events. 2, the temporal information of the topically similar posts always appear around the same time.

A. Data Set

We use a Weibo (similar to Twitter in China) dataset to evaluate our model. The original dataset contains more than 100000 users each one have more than 100000 followees in China. The users were obtained by a set of seed users for tracing their verified follower/followee recursively. Because this data is huge, we sampled part of the data from Jun 13. 2014 to Jun 23. 2014 to evaluate the **precision** and **recall** of our model with the baselines. After preprocessing, we obtained the final dataset with more than 300,000 tweets one day and the number of the words is 199,313.

B. Parameter Setting

In this section, we show the parameters setting of our model. Empirically, a $\lambda_{\theta B}$ suitable for text documents is between 0.05 and 0.1 [17]. In our experiments, we fix our $\lambda_{\theta B} = 0.05$ and set

$$p(w|\theta_B) = \frac{\sum_d n(w,d))}{\sum_w \sum_d n(w,d)} \quad (15)$$

where $p(w|\theta_B)$ denotes the background topic distributed over the vocabulary, and $n(w,d)$ represents the frequency of word w in the document d. The parameter $p(t_d|\theta_B)$ in our algorithm is a constant, means that the background topic appear in the streaming dataset with the same intensity all the time.

C. Ground Truth Generation

To compare the precision of our model with other alternative models, we perform the effectiveness evaluation in our experiments. For our model and PLSA, we get the result of time series data for a number of topics. We apply the same method explained in the bursty detection module to detect the bursty events. We rank the obtained bursty events by the number of tweets (or words in the case of PLSA model, or the keywords in each cluster for TM) assigned to the topics, and take the top-K events from each model. Since no ground

TABLE I: Precision at K for the three models

Method	P@3	P@5	P@10
TM	1.000	0.800	0.500
PLSA	1.000	0.800	0.500
BEE	**1.000**	**0.800**	**0.600**

TABLE II: Recall at K for the three models

Method	P@3	P@5	P@10
TM	0.667	0.750	**0.800**
PLSA	0.667	0.750	0.733
BEE	**1.000**	**0.875**	0.733

truth is available about all the "relevant" events, we manually check the events detected by the three models with the real-world events in the news. As the same idea in [3], we labeled the event to be a bursty event if its number of relevant tweets in our datasets at least doubled in the future time window scope from the first detected time point. In these experiments, we define the **precision** and **recall** as following:

$$\textbf{Precision} = \frac{a}{b}; \tag{16}$$

$$\textbf{Recall} = \frac{a}{c}; \tag{17}$$

where a is the number of the bursty events detected and they are also the real-world events. b is the number of the bursty events detected by the same algorithm. c is the number of the bursty events detected by all the three algorithms and they are the real-world events.

D. Evaluation

In this section, we show the effectiveness evaluation of the three models, namely, PLSA, TwitterMonitor and our BEE. For each algorithm, we set the number of topics (clusters) K to 30 to detect the bursty events from Jun 13. 2014 to Jun 15. 2014. For our BEE, we revise the parameters incorporated with one day of data.

1) Effectiveness: Table I and Table II show the comparison between these models in terms of the precision and recall of the top-K results. Overall, our algorithm can outperform all the other methods in the **precision** for $K = 10$. For $K < 10$, the three models perform same. But in the **recall**, the TM performs the best followed by our model for $K = 10$. We find that it is because the TM detected lots of bursty keywords for its grouping module, while our model ignores the unimportant bursty information.

2) Case Study and Discussions: In this section, we show some sample results from our experiments and discuss some case studies that illustrate the advantages of our model. First, we show the top-5 bursty events from Jun 13. 2014 to Jun 15. 2014 discovered by our model in Table III. As we see, all the events detected by our model are meaningful except that the last event is about an advertising. The events detected range from the football game of the World Cup to the China's anti-corruption. An example is the event World Cup, people focus on the event of opening ceremony of the World Cup

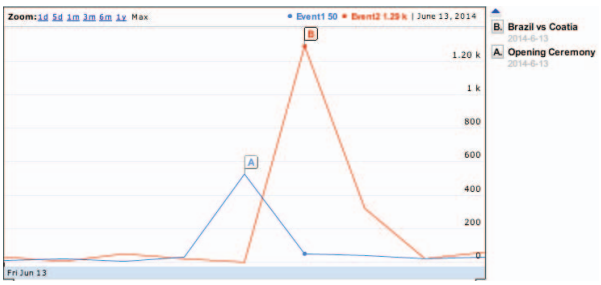

Fig. 3: Intensity of events

Brazil 2014, during 3:00 to 4:00 (UTC+08). While, during 4:00 to 6:00 (UTC+08), the focus drifts to the match between Brazil and Croatia. And the topic of these two events merged together. Our BEE can distinguish the two events based on the different temporal information. But, in the TM and PLSA, they can not distinguish them. We find that, they can not utilize the different temporal information of events to group keywords. From the results, we can see that the topically similar contents appear around the same time and our results also support our hypothesis.

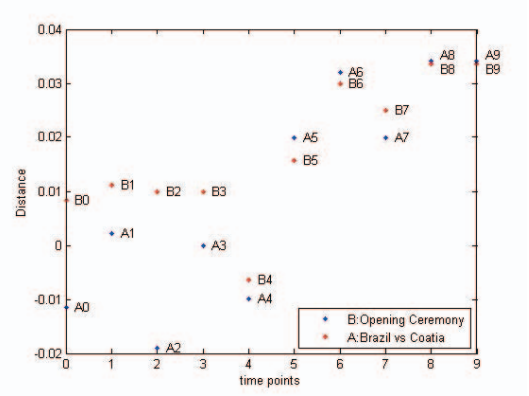

Fig. 4: Topic drifting of *Opening Ceremony, Brazil vs Coatia*

In the Fig. 3, we show the intensity of the events on Opening Ceremony and the match (Brazil vs Croatia) of our model. We can see that with modeling the temporal information, BEE can depict burst characteristics well for analysis.

Fig.4 show the topic drifting of events *Opening Ceremony* and *Brazil vs Coatia*. In this Figure, Ai denotes as the topic of the event of match between Brazil and Coatia at $i-th$ time point, Bi denotes as the topic of the event of the Opening Ceremony of the World Cup Brazil 2014 at $i-th$ time point. The distance between topics represent the dissimilarities of each pair topics calculated by the multidimensional scaling algorithm. The larger the distance is, the more different the two topics are. From the Fig. 4, we can see that the topic of event *Opening Ceremony* drifted to the topic of *Brazil vs Coatia* over time gradually. After $4th$ time point, their topic merged together. BEE can depict the topic drifting well.

TABLE III: The Top-5 Bursty Events Detected by BEE

First Detected Time	Top Words	Label
03:00,Jun 13	开幕式(Opening ceremony),路易斯(Louis), 国歌(national anthem),圣保罗(Sao Paulo) 克罗地亚队(Croatia),大卫(David),当地时间(Local time),揭幕(openere)	Opening Ceremony
05:00,Jun 13	巴西(Brazil),乌龙球(Own goal),世界杯(World Cup),克罗地亚(Croatia) 点球(Penalties),俱乐部(club),内马尔(Neymar), 进球(Goal)	Football Game
06:00,Jun 13	巴西(Brazil),西村(Yuichi),扳平(Equalize),罗纳尔多(Ronaldo) 打进(Goal),主裁判(Chief judge), 补时(Added time), 禁区(Forbidden zone)	Football Game
19:00,Jun 14	青海(Qinghai Province),省委(Provincial),十八大(18th CPC National Congress) 副书记(deputy secretary),首位(The first),痴迷(obsession),落马(arrested) 社科院(Chinese Academy of Social Sciences)	China's anti-corruption
-	N/A	Advertising

V. Conclusion

In this paper, we studied the problem of online bursty event detection from microblog. Because existing works on bursty event detection may not be suitable for our problem, we proposed a new incremental topic model namely BEE that model the temporal information of events in posts streaming. BEE can update the parameters using incremental methods for tracking the topic drifting. We compared our BEE with several baselines on a real Weibo (Chinese Twitter) dataset. Experiments show that our algorithm can outperform the baselines for detecting the meaningful bursty events. We also used a case study to illustrate the effectiveness of the temporal factor for bursty event detection and how well BEE can track the topic drifting.

The limitation of the current method is that the number of topics is predetermined. We also plan to look into methods that update the number of topics along with the timeline. To model the events in microblog well, we plan to study a supervised topic model based on our BEE with the hashtag of microblog. And to handle with the larger dataset, the direction of topic model on Spark [21], a distributed in-memory computing platform, is also our plan to study.

VI. Acknowledgments

This work is supported by China MOST project (No. 2012BAH46B04), NSFC Program (No. 61472022), and SKLSDE-2014ZX-04.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[2] J. L. Boyd-Graber and D. M. Blei. Syntactic topic models. In *Advances in neural information processing systems*, pages 185–192, 2009.

[3] Y. Chen, H. Amiri, Z. Li, and T.-S. Chua. Emerging topic detection for organizations from microblogs. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 43–52. ACM, 2013.

[4] T.-C. Chou and M. C. Chen. Using incremental plsi for threshold-resilient online event analysis. *Knowledge and Data Engineering, IEEE Transactions on*, 20(3):289–299, 2008.

[5] Q. Diao, J. Jiang, F. Zhu, and E.-P. Lim. Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 536–544. ACM, 2012.

[6] A. Gruber, Y. Weiss, and M. Rosen-Zvi. Hidden topic markov models. In *International Conference on Artificial Intelligence and Statistics*, pages 163–170, 2007.

[7] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[8] L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[9] P. Lee, L. V. Lakshmanan, and E. E. Milios. Incremental cluster evolution tracking from highly dynamic network data. In *Data Engineering (ICDE), 30th International Conference on*, pages 3–14. IEEE, 2014.

[10] C. Li, A. Sun, and A. Datta. Twevent: segment-based event detection from tweets. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 155–164. ACM, 2012.

[11] R. Li, K. H. Lei, R. Khadiwala, and K.-C. Chang. Tedas: A twitter-based event detection and analysis system. In *Data Engineering (ICDE), 28th International Conference*, pages 1273–1276. IEEE, 2012.

[12] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 227–236. ACM, 2011.

[13] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the SIGMOD International Conference on Management of data*, pages 1155–1158. ACM, 2010.

[14] S. Phuvipadawat and T. Murata. Breaking news detection and tracking in twitter. In *Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 3, pages 120–123. IEEE, 2010.

[15] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[16] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.

[17] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 784–793. ACM, 2007.

[18] J. Weng and B.-S. Lee. Event detection in twitter. In *ICWSM*, 2011.

[19] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang. Topicsketch: Real-time bursty topic detection from twitter. 2013.

[20] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1445–1456. International World Wide Web Conferences Steering Committee, 2013.

[21] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 2–2. USENIX Association, 2012.

[22] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. Springer, 2011.