Exploiting Temporal Divergence of Topic Distributions for Event Detection

Rongda Zhu, Aston Zhang, Jian Peng, Chengxiang Zhai

University of Illinois at Urbana-Champaign
Urbana, IL USA
Email: {rzhu4, lzhang74, jianpeng, czhai}@illinois.edu

***Abstract*—In a collection of documents, such as news articles or tweets, various events take place over time. The event detection problem aims at discovering significant events that have not been mentioned before the detection time. When these events occur, we observe that topic distributions of documents will diverge notably. However, event detection from such divergence may be hampered by noises. In this paper, we propose $TopicDiver$, a novel method to address the event detection problem. $TopicDiver$ models topic distributions of documents over time and filters out noises while capturing the useful divergence between such distributions. The direct exploitation of topic distribution over time sets our work apart from existing studies on event detection. We conduct comprehensive experiments under different settings on news and Twitter data. The experimental results demonstrate that $TopicDiver$ outperforms the baseline models in the measures for accuracy across various settings.**

***Keywords*-Event Detection, Topic Distribution, Text Stream, Time**

I. Introduction

In this big data era with overwhelming online text information, the problem of event detection from text data has received increasing attention. The discovery of novel and important events can greatly help one better understand the data. For example, event detection on scientific literature can be useful to help new researchers understand the evolving research themes over time [1]. With the increasing popularity of social media, event detection on Twitter has been widely used for seizing and analyzing the social trends and public interests [2], and even for faster earthquake detection [3].

Existing event detection methods can be generally grouped into *document-pivot* and *feature-pivot*. *Document-pivot* methods try to cluster the documents and extract event features from these clusters. For example, the U-MASS system [4] achieving the best performance in several topic detection and tracking (TDT) competitions uses *term frequency–inverse document frequency* (TF-IDF) weight vectors to represent the documents, and assigns the incoming document to the nearest cluster if the similarity is above a predefined threshold. Otherwise, the incoming document is identified as a new event if it is different enough from any previous document. The efficiency of the UMASS system is further improved by locality-sensitive hashing (LSH) [5], making it scalable for social media like Twitter.

On the other hand, *feature-pivot* methods focus on detecting the statistical patterns of the corpus and get event features from these patterns. For example, Fung *et al*. [6] proposed to model the term frequency using a binomial distribution. When the statistics of the distribution has changed, the bursty features are detected as a set of words. Finally, an event is characterized by a set of co-occuring bursty features. In [7], the authors further extended this work to an event hierarchy construction problem, where the documents are clustered into an event hierarchy based on their bursty features. Frequency-domain technique is introduced in [8], where the authors utilized *Discrete Fourier Transform* (DFT) to extract the bursty features from term frequency. By its nature, their method has an advantage on discriminating periodic and aperiodic events.

One of the traits of social media such as Twitter is that it will generate very high volume of meaningless "babbles" [5]. Weng *et al*. [9] used the clustering of wavelet signals generated from term frequency and filtered the trivial terms by the autocorrelation of their wavelet signals. Another straightforward method to overcome the babbles in Twitter was introduced in [10]. The authors used the hash tag *#breakingnews* to pick the breaking news posts. To group the similar Tweets which are sometimes short-length, they boosted the weight of proper nouns in TF-IDF weighting. Based on the temporal and geographical features of social media such as Twitter, there are also works proposing location or time-based topics to capture the events [11] over multiple dimensions.

Events detected by these *feature-pivot* methods are often described by a set of keywords [2], [6]. This inspires us to build our method on top of topic models, which are proved to be extremely successful in discovering the hidden "topics" from text corpus and characterizing these topics by keywords. The input of a typical topic model is the text corpus, and the output includes:

- *Word distribution* of topics $p(w|z)$: given a topic z, its probability of generating a word w in vocabulary. We have $\sum_{i=1}^{|V|} p(w|z) = 1$.
- *Topic distribution* of documents $p(z|d)$: given a document d in corpus, the probability it's about a topic z. We have $\sum_{k=1}^{K} p(z_k|d) = 1$.

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

For example, for a specific topic z_k about computer industry, the words with the highest probability may be $P(\text{“}computer\text{”}|z_k) = 0.05, P(\text{“}software\text{”}|z_k) = 0.04$ and $P(\text{“}technology\text{”}|z_k) = 0.02$. If a specific document d is about computer industry, then z_k may be the most relevant topic with the highest $P(z_k|d)$.

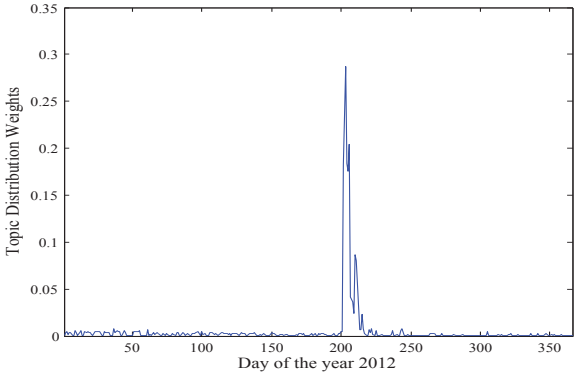
Static topic models such as probabilistic latent semantic analysis (PLSI) [12] and latent dirichlet allocation (LDA) [13] have been widely applied to many different tasks with intrinsic relation to event detection, however, it still remains an open challenge how to effectively leverage them into this problem. Alsumait *et al.* [14] proposed an online variant of LDA with applications to event detection. The topic model is learned in an online fashion, and the parameters of the last model are transferred as priors to the current model. A new topic is identified when the word distribution is different enough. Lau *et al.* [15] further extended this model to a dynamic vocabulary. They also focused on the changes of the word distribution of topics to detect events.

Although there exist various topic model-based methods on event detection, most of them look into the divergence of *word distributions* of topics to identify events. The *topic distributions* of documents, which can be viewed as the strength or coverage of topics, are often neglected in these methods. Is such neglected information helpful for event detection? We begin with a motivating example where a real event emerged.

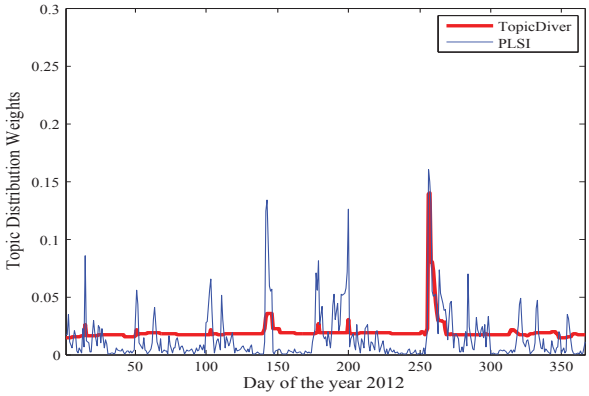
Motivating Example. On July 20, 2012, a mass shooting occurred in a theater in Aurora, Colorado[1]. The topic model PLSI [12] is applied on the CNN dataset with details deferred in Section IV. The daily topic distribution corresponding to mass shooting around the event date is exhibited in Figure 1(a). The topic distribution on a specific day is the average of the topic distributions of all documents on the day. The top words associated with this topic include “Aurora”, “gunman”, “theater”, and “victim”. The peak in Figure 1(a) corresponds to the exact date of the event. As the coverage of the event lingers a few days after the event date, this peak gradually disappears as the coverage goes down. Therefore, the divergence of topic distributions between adjacent time stamps may precisely indicate the occurrence of a new event.

Challenge. We highlight that capturing such temporal divergence of topic distributions is challenging. To illustrate, Figure 1(b) depicts the daily distribution of the topic on the situation in Libya over time. The thin blue line denotes the topic distribution of PLSI generated the same way as the mass shooting topic, where numerous peaks can be observed. However, most of these peaks are not related to events that are noteworthy. According to our manual annotation, only one event (the peak with a circle in Figure 1(b)) corresponds to the topic of Libya’s situation. On most of the other

[1]https://en.wikipedia.org/wiki/2012_Aurora_shooting



(a) Topic on Aurora Shooting



(b) Topic on Libya Situations

Figure 1. Temporal Distribution for Two Topics

“peaks” in the figure, the divergence is caused by updates of status, follow-ups of events, or general discussions. It is interesting to note that news stories are not necessarily always discussing new events that we want to detect.

Different from the mass shooting topic that is about a single emergency, the Libya’s situation topic is broader with multiple aspects that evolve over time. Therefore, the divergence of the Libya’s situation topic distributions over adjacent time may be affected by other non-event factors. We refer to such non-event divergence as noises since it hampers the detection of real events. Such noise is actually very common in the detection of important events, as most of the significant events involve effects in multiple aspects, cause different follow-ups that last a long time period.

Our Contribution. We take the initiative towards exploiting the hitherto-undiscovered temporal divergence of topic distributions for event detection. Intuitively, when an event takes place, there will be a lot of documents discussing it. Therefore, the average topic distribution of the documents on the topic corresponding to the event will go up. The more significant the event is, the larger this increase should be. Compared with the word distributions of topics which are

more complicated and affected by more factors, the topic distribution serves as a more straightforward sign of the change in corpus themes. While quantifying the distance of word distributions is always involved with complex measure such as KL-divergence, another advantage of topic distribution is that the difference is much simpler and easier to use, as we will show in later sections.

In this work, we propose a novel event detection method, *TopicDiver*, based on the divergence of topic distributions over time. We first build *TopicDiver* on top of a mixture model similar to PLSI, but on time-stamped text streams. To overcome the noise challenge, we then leverage the longitudinal regularization on the difference of adjacent topic distributions to effectively capture the temporal divergence of topic distributions caused by real events. In sharp contrast to multiple noisy peaks shown in the thin blue line in Figure 1(b), the thick red line is the corresponding Libya's situation topic distribution generated by *TopicDiver*. We can see that *TopicDiver* effectively smooths out noises and captures temporal divergence of topic distributions corresponding to the real event.

We evaluate *TopicDiver* with various datasets from newsire and social media. The results show that *TopicDiver* outperforms the state-of-the-art methods consistently, especially at the detection of significant events which draw a lot of coverage. The proposed *TopicDiver* is a general model that can be applied to any text data in any natural language for discovering new events. It is also efficient to be applied to large-scale dataset.

II. Problem Formulation

In this section, we will formalize our event detection problem. The input for the event detection problem is a time-stamped text stream, represented by a collection of documents over a set of time stamps. These time stamps can be at any reasonable granularity based on the data. For example, if we want to analyze the scientific literature in computer science over the past few decades, year should be an appropriate time unit here; but if our data is Twitter stream generated at a very high rate, we might use hour as the time unit. The time stamps are denoted by $t_1, t_2, \ldots, t_T$, and the collection of documents published on t_i is denoted by $C_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,N_i}\}$, where $d_{i,l}$ is the l-th document on this time stamp. The input collection is denoted by $D = \{C_1, \ldots, C_T\}$.

The output should be a set of detected $events$, where each event is denoted by one of the following two: noitemsep

- A $first\ story\ d_{t,k}$, the first document discussing the event in D.
- A set of words $\{w_1, w_2, \ldots w_M\}$ that can describe the event.

As we will show later, *TopicDiver* can be conveniently adapted into both settings. We have two settings of event detection: noitemsep

- Retrospective: we have the entire collection D available and want to detect events on all the time stamps $\{t_1, \ldots, t_T\}$.
- Online: we have the documents up to time t_i, *i.e.* $C_1, \ldots, C_i$ available when the detection time is t_i.

From the above description, we can see that if the time granularity is fine enough, both settings are at the document level, *i.e.* each C_i has only one document and we determine whether it is a new event right on its published time.

III. Proposed Method

In this section, we will describe our proposed method, *TopicDiver*. We will start with PLSI, one of the most popular topic models, and then build *TopicDiver* on top of it.

A. Probabilistic Latent Semantic Indexing (PLSI)

PLSI is a widely-used model analyzing the hidden topics of text corpus, featured by latent variables. Specifically, given a co-occurrence of a word, document pair (w, d), the probability of the pair is modeled as the mixture of K different topics:

$$P(w, d) = P(d) \sum_{k=1}^{K} P(w|z_k)P(z_k|d),$$

where each z_k is a hidden topic, and $P(w|z_k)$, $P(z_k|d)$ are what we refer as word distribution of topic z_k and topic distribution of document d respectively.

While most existing methods look into the divergence of $P(w|z_k)$ at adjacent time stamps to detect events, we use $P(z_k|d)$ over time. For example, when generating the curve in Figure 1(a) for retrospective event detection, we first run PLSI on the whole corpus to generate $P(z_k|d)$ for all document d. For each time stamp t_i, we compute $P(z_k|t_i)$ as the average of $P(z_k|d_{i,l})$ and plot it against time. We then use the criterion $P(z_k|t_i) > 3P(z_k|t_{i-1})$ to easily check if there is an event on t_i corresponding to topic z_k.

B. TopicDiver: A Longitudinal Regularized Mixture Model

We now describe $TopicDiver$ as a two-step extension of PLSI, starting from a mixture model for text streams, and then introducing a longitudinal regularization.

1) A Mixture Model for Text Streams: Recall that the input of the problem is a collection D. Since our method utilizes the topic distributions over time, we concatenate all the documents published on the same time stamp to form a $super\ document$. For example, documents in C_i will form a $super\ document\ S_i$. Obviously, when the time granularity is fine enough, the $super\ documents\ S_i$ are just the documents d_i. The vocabulary of S_i is denoted by V_i, and the vocabulary of the collection is $V = \cup_{i=1}^{T} V_i$. We use $f(w, d)$ to denote the count of a certain word w in a certain document or $super\ document\ d$.

Given a collection of *super documents* $S = \{S_1, S_2, \ldots, S_T\}$, the log-likelihood of S is given by the mixture model:

$$\log P(S) = \sum_{i=1}^{T} \log P(S_i) = \sum_{i=1}^{T} \sum_{j=1}^{|V_i|} f(w_j, S_i) \log P(w_j|t_i), \tag{1}$$

where $P(w_j|t_i)$ is denoted as the mixture of K topics $\{z_1, z_2, \ldots, z_K\}$,

$$P(w_j|t_i) = \sum_{k=1}^{K} P(w_j|z_k) P(z_k|t_i).$$

We use $\boldsymbol{\beta}$ to denote the matrix of word distributions of topics, *i.e.* $\beta_{k,j} = P(w_j|z_k)$ and $\boldsymbol{\theta}$ to denote the matrix of topic distributions over time, *i.e.* $\theta_{i,k} = P(z_k|t_i)$. The divergence of $\boldsymbol{\theta}$ over time is the key of $TopicDiver$.

2) Longitudinal Regularization: From [12], we know that the direct maximization of document likelihood in (1) is the process of PLSI on *superdocuments*. However, PLSI deals with static vocabulary with no temporal information considered. Moreover, directly using the topic distributions generated by a conventional topic model will bring much noise for precise event detection, as we will show in the experiments in Section IV. Inspired by the idea of *fused lasso* [16], we apply ℓ-1 regularization on the successive differences of topic distributions. Formally, our framework is given by

$$\begin{aligned}(\boldsymbol{\theta}^*, \boldsymbol{\beta}^*) = \arg\min_{\boldsymbol{\theta}, \boldsymbol{\beta}} & - \sum_{i=1}^{T} \sum_{j=1}^{|V_i|} f(w_j, S_i) \log \sum_{k=1}^{K} \theta_{i,k} \beta_{k,j} \\ & + \lambda \sum_{i=2}^{T} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}\|_1, \\ \text{subject to} \quad & \sum_{k=1}^{K} \theta_{i,k} = 1, \quad i = 1, \ldots, T.\end{aligned} \tag{2}$$

where $\boldsymbol{\theta}_i$ denotes the i-th row of $\boldsymbol{\theta}$.

From (2), we know that the regularization parameter λ is indicating the regularization strength. When λ goes to infinity, we will allow no divergence and $\boldsymbol{\theta}_i$ will be constant along the time; when λ is zero, our framework will become conventional PLSI on the *super documents*.

To sum up, the key differences of *TopicDiver* and conventional PLSI are two-fold:

- We introduce the time variable and apply mixture model on time-stamped text streams instead of documents.
- We add longitudinal regularization on topic distributions of adjacent time stamps.

C. Optimization Algorithm

Now we describe our algorithm to solve the optimization problem in (2). We discuss retrospective and online settings respectively. In retrospective setting, the complete collection D is available, so the vocabulary V is also known. We can directly set $V_i = V$ and use a coordinate descent over $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$. Note that for the constraint, we introduce $\widetilde{\boldsymbol{\theta}}$ and let

$$\theta_{i,k} = e^{\widetilde{\theta}_{i,k}} / \sum_{k'=1}^{K} e^{\widetilde{\theta}_{i,k'}}, \quad i = 1, \ldots, T, \quad k = 1, \ldots, K.$$

We use L to denote the objective function in (2). Then the gradient is given by:

$$\begin{aligned}\frac{\partial L}{\partial \widetilde{\theta}_{i,k}} = & - \sum_{j=1}^{V} f(w_j, S_i) \left[\frac{e^{\widetilde{\theta}_{i,k}} \beta_{k,j}}{\sum_{k'=1}^{K} e^{\widetilde{\theta}_{i,k'}} \beta_{k',j}} - \frac{e^{\widetilde{\theta}_{i,k}}}{\sum_{k'=1}^{K} e^{\widetilde{\theta}_{i,k'}}} \right] \\ & + \lambda \mathbf{sign}(\widetilde{\theta}_{i,k} - \widetilde{\theta}_{i-1,k}), \\ \frac{\partial L}{\partial \beta_{k,j}} = & - \sum_{i=1}^{T} f(w_j, S_i) \frac{e^{\widetilde{\theta}_{i,k}}}{\sum_{k'=1}^{K} e^{\widetilde{\theta}_{i,k'}} \beta_{k',j}}.\end{aligned}$$

We update $\widetilde{\boldsymbol{\theta}}$ and $\boldsymbol{\beta}$ as following:

$$\widetilde{\boldsymbol{\theta}}^{(n+1)} = \widetilde{\boldsymbol{\theta}}^{(n)} - \gamma_1 \frac{\partial L}{\partial \widetilde{\boldsymbol{\theta}}}, \quad \boldsymbol{\beta}^{(n+1)} = \boldsymbol{\beta}^{(n)} - \gamma_2 \frac{\partial L}{\partial \boldsymbol{\beta}}. \tag{3}$$

The algorithm stops when L converges.

In the online setting, the vocabulary evolves over time. Therefore, we need to fold in the new words and documents in a streaming fashion. We use the method in [17] folding in new words and documents. Specifically, we run a topic model on the first *super document* S_1 to get $\boldsymbol{\beta}$ and $\boldsymbol{\theta}_1$. After finishing detection on t_{i-1}, we do the following steps for detection on t_i: noitemsep,nolistsep

1) Fold in new documents. For each $d_{i,l}$ in C_i we initialize all $P(z_k|d_{i,l})$ randomly. We adopt the EM algorithm to compute

$$P(z_k|w_j, d_{i,l}) = \frac{P(w_j|z_k) P(z_k|d_{i,l})}{\sum_{k'=1}^{K} P(w_j|z_{k'}) P(z_{k'}|d_{i,l})}, \tag{4}$$

$$P(d_{i,l}|z_k) = \frac{\sum_{j=1}^{|V_i|} f(w_j, d_{i,l}) P(z_k|w_j, d_{i,l})}{\sum_{l=1}^{N_i} \sum_{j=1}^{|V_i|} f(w_j, d_{i,l}) P(z_k|w, d_{i,l})}. \tag{5}$$

2) Fold in new words. We use w_{new} to denote the new words in V_i that are not in $V_1, \ldots, V_{i-1}$, and compute

$$P(z_k|w_{new}, d_{i,l}) = \frac{P(d_{i,l}|z_k) P(z_k|w_{new})}{\sum_{k'=1}^{K} P(d_{i,l}|z_{k'}) P(z_{k'}|w_{new})}, \tag{6}$$

$$P(z_k|w_{new}) = \frac{\sum_{l=1}^{N_i} f(w_{new}, d_{i,l}) P(z_k|w_{new}, d_{i,l})}{\sum_{l=1}^{N_i} f(w_{new}, d_{i,l})}. \tag{7}$$

Then we will use (3) to do coordinate descent.

After we get $\boldsymbol{\beta}^*$ and $\boldsymbol{\theta}^*$, we can use the divergence in $\boldsymbol{\theta}^*_{i-1}$ and $\boldsymbol{\theta}^*_i$. We will use the simple but effective rule $\theta^*_{i,k} > 3\theta^*_{i-1,k}$ to determine if there is enough divergence to indicate

Algorithm 1: *TopicDiver*: Online Event Detection from Text Streams

Input: the text corpus D with time stamps $\{t_1, \ldots, t_T\}$, number of topics K
Output: A set of detected events, each featured by a set of keywords and top document(s).
Initialize $\boldsymbol{\beta}$ and $\boldsymbol{\theta}_1$ from topic model on S_1, the set of detected event documents $\mathcal{E} \leftarrow \emptyset$, the set of detected event keyword sets $\mathcal{W} \leftarrow \emptyset$
for $i = 2$ ***to*** T **do**
  Concatenate all documents from D on t_i to get S_i.
  for *each document* $d_{i,l}$ **do**
    Fold in $d_{i,l}$ using 4 and 5
  for *each new word* w_{new} **do**
    Fold in w_{new} using 6 and 7
  Gradient Descent using 3
  for *each topic* z_k **do**
    if $\theta^*_{i,k} > 3\theta^*_{i-1,k}$ **then**
      Find top documents $d_{i,k}$ related to z_k by 8, and top terms $w_{i,k}$ by $P(w|z_k)$
      $\mathcal{E} = \mathcal{E} \cup \{d_{i,k}\}$
      $\mathcal{W} = \mathcal{W} \cup \{w_{i,k}\}$
return $\mathcal{E}$, $\mathcal{W}$

an event in topic k emerged on t_i. When such divergence is detected, the top words associated with the topic can be directly used to describe the event.

D. Document Ranking

We have got the description keywords for events from the diverging topic, and we now use the other output setting, *i.e.* select the event documents in the collection as output. Since we have already got the time stamps when topic distributions diverge, we only need to rank the documents on this time stamp. The ranking function for a document is defined as

$$r(d, t_i, z_k) = P(z_k|d) \cdot e^{-\eta n_d}, \tag{8}$$

where $P(z_k|d)$ denotes how relevant the document d is to the k-th topic, n_d is the temporal order of d in C_i, and $\eta = 0.5$ is the decaying rate penalizing later documents. After ranking all the documents, we set a threshold value to identify all the first story documents.

We show the outline of the online version of $TopicDiver$ algorithm in Algorithm 1.

IV. Experiments

In this section, we evaluate $TopicDiver$ on data sets from news articles and social media. For all the quantitative evaluation metrics, we use first story documents as output. We also use keywords to qualitatively illustrate the example events we have detected. We show that $TopicDiver$ outperforms other state-of-the-art methods, and is especially good at detecting significant events.

A. Datasets

We use three datasets for our experiments, two from newswire and one from social media. We first testify our algorithm on news datasets, which are standard TDT5 dataset and CNN TV transcripts before moving to the social media dataset from Twitter.

Standard TDT5 Dataset: The standard TDT5 dataset is the benchmark dataset widely used in several TDT contests. It consists of news articles from various news media in multiple languages. We will use only the English part, containing 126 events labeled with first story in 221306 documents spanning 183 days from April to September 2003, with a vocabulary size of 87790 after preprocessing. Each day is used as a time stamp.

CNN TV Transcripts: We collect transcripts of several CNN TV shows from 2009 to 2012. Transcripts are the on-screen text during programs, which are good description of the events covered by the program. We manually label events with the transcripts of the first programs covering them. There are 33593 documents and 50 events in total, with the vocabulary size 28670. The time stamp for this dataset is also day.

Twitter Dataset: We collected 26 millions Tweets with over 180 million tokens from March 1st to 20th, 2016, using Apollo System [2]. The total size of the dataset is 98.9GB. Since tweets are generated at a very high rate, we use hour as our time stamp to match the pace. Even though hashtags and special characters such as at signs would be potential indicators of the Tweet content [10], we remove all the hashtags and at signs in the tweets to maintain the generality of our method. We also only select the tweets in English.

B. Evaluation Metrics

Due to the different natures of the datasets, we will now introduce the evaluation metrics for newswire and Twitter data respectively.

Newswire Datasets. For the TDT5 dataset, we follow the ***official*** TDT evaluation plan [4] using *minimal normalized cost*, which is the most popular metric for detection problems. For CNN data, we use both *minimal normalized cost* and F_1 score. We first introduce the basic measures: noitemsep

- $Precision$: fraction of detected documents that are events.
- $Recall$: fraction of events that are detected.
- $False\ Alarm$ *(FA)*: fraction of non-event documents that are detected as events.
- $Miss$: fraction of events that are not detected.
- $F_1 = 2 \cdot Precision \cdot Recall/(Precision + Recall)$.

We now introduce *minimal normalized cost*. First, we define *detection cost* C_{det} as

$$C_{det} = Miss \cdot C_{miss} \cdot P_{target} + FA \cdot C_{FA} \cdot P_{non\text{-}target},$$

[2]http://apollo3.cs.illinois.edu/

According to the official TDT evaluation plan [4], we set $C_{miss} = 1$ as the cost of missing an event; $C_{FA} = 0.1$ as the cost of detecting a non-event document as an event, $P_{target} = 0.02$ and $P_{non\text{-}target} = 0.98$ as the prior probability of an event document in the corpus. We can easily see that $C_1 = C_{FA} \cdot P_{non\text{-}target}$ and $C_2 = C_{miss} \cdot P_{target}$ are the costs of declaring all documents events and non-events respectively. The normalized detection cost is defined as

$$C_{norm} = \frac{C_{det}}{\min\{C_1, C_2\}}.$$

Finally, different parameter values will lead to different miss and false alarm values. In [4], the authors do a parameter sweep on the threshold. In our case, λ is an important parameter controlling the strength of regularization, and thus the number of events detected. Therefore, we use grid search to determine the best value of λ minimizing C_{norm}. The *minimal normalized cost* C_{min} is the minimum of C_{norm}.

Twitter Dataset. The evaluation metrics for Twitter dataset is different. Given the vast volume and rapid generating rate of Twitter data, it is not practical either to label all the tweets or to choose an event and find the first tweet mentioning it. We evaluate methods on the tweets detected instead of the whole collection, which is the method used in many other works [2], [5], [9]. For evaluation on Twitter dataset, we use $precision$, which is the fraction of selected tweets related to events (not necessarily the earliest), and $recall$, which is now the number of unique events detected on a daily basis [9]. Since Twitter data is often overwhelmed with noises potentially undermining the event detection, we will also use number of detections to check if the model can generate both precise and concise results.

C. Experiment Design

We now introduce and verify the design of our experiment. We want to test and show the following aspects through our experiments:

The effect of longitudinal regularization. First of all, recall that PLSI is a special case of $TopicDiver$ where $\lambda = 0$. Therefore, we want to compare $TopicDiver$ and PLSI to demonstrate the effect of our longitudinal regularization. Since PLSI is a static topic model and online variants are not directly related to $TopicDiver$, we only compare $TopicDiver$ and PLSI on retrospective event detection of newswire datasets to see the effect of longitudinal regularization alone.

The efficacy of *TopicDiver* on newswire datasets. We want to testify the efficacy of *TopicDiver* on newswire datasets. In newswire datasets, the documents are well-written news articles in formal language. Note that the events labeled in CNN dataset are mostly significant ones with extensive coverage, and the events in TDT5 dataset also include some less important ones with less and short coverage. Since the online setting is more challenging and important in real application, we will only use this setting and show the comparison between *TopicDiver* with the baselines including the UMASS system [4] which performed best in several TDT competitions, and the improved algorithm based on LSH with variance reduction, proposed in [5].

The efficacy of *TopicDiver* on social media. As we have mentioned earlier, social media is very different from newswire data, with a rapid generating rate and a lot more informal language, meaningless babbles and personal conversation. Due to the rapid pace and timeliness of social media, retrospective event detection on Twitter is far less meaningful. Therefore, only online event detection is conducted on Twitter data. Since the UMASS system is not designed to work on web scale, we replace it with IPLSI introduced in [17]. By grid search, we set number of topics 80 for *TopicDiver* and IPLSI.

D. Experimental Results

We now show the experiment results to testify the efficacy of longitudinal regularization and *TopicDiver*. To reduce the variance, all results shown are the mean values of ten runs of the systems. First of all, we look into the comparison between *TopicDiver* and PLSI on retrospective event detection. From Table I, we can observe that PLSI suffers from the noises and $TopicDiver$ improves precision by and false alarm, thus F_1 and C_{min} greatly. The advantage of *TopicDiver* is especially remarkable on CNN data, which is expected because the events there are more important, and longitudinal regularization can effectively filter out noises without hurting the more significant divergence points caused by real events. This further validates our idea of adding longitudinal regularization on top of PLSI.

Secondly, we verify *TopicDiver* on online event detection from news data. From Table II, we can see that *TopicDiver* again has a remarkable advantage on CNN data. This is also a demonstration of the effect of the longitudinal regularization. We are achieving comparable performance with the baselines on TDT5 data, with better false alarm and a slightly higher miss. This is because that the events in TDT5 dataset contain some minor ones with less coverage, and the divergence they cause can be smoothed out mistakenly by our regularization. However, for the more important events, we are actually still better than the baselines.

For qualitative evaluation, we list all the labeled and detected events throughout the year 2012 in the CNN dataset, and ten of the detected events in the TDT5 dataset for comparison, in Table III and Table IV. The dates are from the divergence points we have detected, and the keywords are from the top words of the topic of which the temporal distribution is diverging. We can see that all the events in the CNN dataset are important ones which will attract most of the coverage at the time of its emergence. In the contrary, the events in the TDT5 data may be less significant.

Table I
RETROSPECTIVE EVENT DETECTION ON NEWS DATA. A SMALLER $\mathbf{C_{min}}$ OR A LARGER $\mathbf{F_1}$ IS BETTER.

	TDT5			CNN					
Method	False Alarm	Miss	$\mathbf{C_{min}}$	Precision	Recall	$\mathbf{F_1}$	False Alarm	Miss	$\mathbf{C_{min}}$
PLSI	0.026	0.468	0.595	0.041	0.740	0.078	0.026	0.260	0.386
TopicDiver	0.011	0.492	**0.546**	0.248	0.700	**0.366**	0.003	0.300	**0.316**

Table II
ONLINE EVENT DETECTION ON NEWS DATA. A SMALLER $\mathbf{C_{min}}$ OR A LARGER $\mathbf{F_1}$ IS BETTER.

	TDT5			CNN					
Method	False Alarm	Miss	$\mathbf{C_{min}}$	Precision	Recall	$\mathbf{F_1}$	False Alarm	Miss	$\mathbf{C_{min}}$
UMASS	0.042	0.492	**0.696**	0.132	0.660	0.220	0.007	0.340	0.372
LSH	0.044	0.492	0.707	0.112	0.660	0.191	0.008	0.340	0.379
TopicDiver	0.037	0.524	0.703	0.165	0.700	**0.267**	0.005	0.300	**0.326**

Table III
LABELED EVENTS IN 2012 CNN TRANSCRIPTS

News Event	Date	Keywords
Death of Whitney Houston	Feb 11, 2012	'Whitney', 'Houston', 'death'
Shooting of Trayvon Martin	Feb 27, 2012	'Trayvon', 'Martin', 'Zimmerman'
Jerry Sandusky's Trial	Jun 12, 2012	'Sandusky','child', 'scandal'
Aurora Shooting	Jul 20, 2012	'Aurora', 'Victims', 'Gun'
London Olympics	Jul 28, 2012	'Olympic', 'London', 'medal'
Hurricane Isaac	Aug 21, 2012	'hurricane', 'storm', 'Louisiana'
Benghazi Attack	Sept 11, 2012	'Benghazi', 'attack', 'arm'
Hurricane Sandy	Oct 22, 2012	'flood', 'hurricane', 'storm'
Presidential Election	Nov 6, 2012	'obama', 'election', 'president'
Sandy Hook Shooting	Dec 14, 2012	'shooting', 'connecticut' 'elementary'

Table IV
LABELED EVENTS IN TDT5 DATASET

News Event	Date	Keywords
London Marathon	Apr 13, 2003	'London', 'marathon', 'competition', 'Radcliffe'
Bombing in Riyadh, Saudi Arabia	May 12, 2003	'Riyadh', 'explosion', 'Arabia', 'terrorist'
Hu Jintao meets Bush	Jun 01, 2003	'president','Bush', 'China', 'Korea'
U.S. Helicopter Crashed in Kosovo	Jun 08, 2003	'helicopter', 'Kosovo', 'crash'
Two Britons among terror suspects	Jul 04, 2003	'Abbasu', 'Begg'[3], 'Cuba'
2003 World Swimming Championship	Jul 20, 2003	'swim', 'record', 'champion', 'Thorpe'
Wildfire in Portugal	Aug 09, 2003	'Portugal', 'forest', 'fire', 'flame'
Wu Bangguo visits Manila	Aug 30, 2003	'Chinese', 'Philippines', 'policy'
Earthquake in Japan	Sept 26, 2003	'Hokaido', 'Japan', 'earthquake'
First Nigerian satellite in space	Sept 27, 2003	'Nigerian', 'launch' 'satellite'

Table V
EVENT DETECTION ON TWITTER DATA. A LARGER PRECISION OR RECALL IS BETTER.

Method	No. of Detections	Precision	Recall
IPLSI	752	0.331	**45**
LSH	188	0.601	37
TopicDiver	147	**0.755**	42

Finally, we look at results on Twitter data. From Table V, we observe that IPLSI claims far more events than the other two. This is also expected, because IPLSI is designed as an online variant of PLSI, and it suffers the similar problem with PLSI. With regularization to filter out the noise, *TopicDiver* has a great advantage on precision over IPLSI with only minor loss on recall, and also outperforms LSH remarkably both on precision and recall.

TopicDiver is also efficient in terms of complexity and runtime. For retrospective event detection, the optimization problem of our framework can be easily delivered by a stochastic gradient descent, which is much more efficient than EM algorithm of PLSA. In the online setting, since *TopicDiver* folds in new words and documents incrementally, it's also efficient compared to document based methods such as the UMASS system. In our experiments, *TopicDiver* is comparably efficient with LSH method, faster than IPLSI, and far more efficient than PLSI and the UMASS system.

E. Parameter Setting

As we have mentioned in previous sections, the most important parameters of *TopicDiver* is the regularization

[3]Abbasu and Begg are people names.

Table VI
EFFECT OF REGULARIZATION PARAMETER λ

λ	# Detections	False Alarm	Miss	$\mathbf{C_{norm}}$
0	900	0.026	0.260	0.386
0.02	819	0.023	0.260	0.374
0.05	524	0.014	0.280	0.351
0.1	235	0.006	0.300	0.329
0.2	145	0.003	0.300	**0.316**
0.3	93	0.002	0.340	0.349
0.4	65	0.002	0.420	0.426
0.6	36	0.001	0.700	0.703

parameter λ. Recall that λ in our method controls the regularization strength and thus the number of detected events by our algorithm.

Intuitively, the larger λ is, the more regularization we put onto *TopicDiver* and the more rigorous we are on the events we detect. In this way, we are more likely to detect significant events causing really large changes in topic distributions. On the other hand, we expect more detections including some minor events if λ is smaller. When $\lambda = 0$, our model will become the conventional PLSI model. Therefore, we can flexibly choose the value of λ based on both our information needs and the data.

We show the number of detected events and the detection accuracy on CNN dataset in retrospective mode against the value of lambda in Table VI, with the best value in bold. We can see that the value of λ has a large impact on the detection results, with larger λ causing higher miss and lower false alarm, and vice versa. However, the optimal values may largely depend on the datasets. Therefore, we use grid search in our experiments to determine the best parameter values.

V. CONCLUSION

In this paper, we proposed $TopicDiver$, a novel event detection method. In contrast to existing methods, *TopicDiver* directly exploits divergence of topic distributions over time. By leveraging the longitudinal regularization, $TopicDiver$ effectively smooths out noises and boosts the detection precision, especially for significant events. We demonstrated the efficacy of $TopicDiver$ with extensive experiments on both news and Twitter data.

REFERENCES

[1] J. Kleinberg, "Bursty and hierarchical structure in streams," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 91–101.

[2] 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*, ser. CIKM '12. ACM, pp. 155–164.

[3] 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*, ser. WWW '10. ACM, 2010, pp. 851–860.

[4] J. Allan, V. Lavrenko, D. Malin, and R. Swan, "Detections, bounds, and timelines: Umass and tdt-3," in *IN PROCEEDINGS OF TOPIC DETECTION AND TRACKING WORKSHOP (TDT-3)*, 2000.

[5] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. ACL, 2010, pp. 181–189.

[6] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 181–192.

[7] G. P. C. Fung, J. X. Yu, H. Liu, and P. S. Yu, "Time-dependent event hierarchy construction," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 300–309.

[8] Q. He, K. Chang, and E.-P. Lim, "Analyzing feature trajectories for event detection," in *Proceedings of the 30th ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 207–214.

[9] J. Weng and B.-S. Lee, "Event detection in twitter." *International AAAI Conference on Web and Social Media ICWSM*, vol. 11, pp. 401–408, 2011.

[10] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in twitter," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 3. IEEE, 2010, pp. 120–123.

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

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

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[14] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking," in *2008 8th IEEE international conference on data mining*. IEEE, pp. 3–12.

[15] J. H. Lau, N. Collier, and T. Baldwin, "On-line trend analysis with topic models:\# twitter trends detection topic model online." in *COLING*, 2012, pp. 1519–1534.

[16] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[17] T.-C. Chou and M. C. Chen, "Using incremental plsi for threshold-resilient online event analysis," *IEEE transactions on Knowledge and Data Engineering*, vol. 20, no. 3, pp. 289–299, 2008.