TrioVecEvent: Embedding-Based Online Local Event Detection in Geo-Tagged Tweet Streams

Chao Zhang[1], Liyuan Liu[1], Dongming Lei[1], Quan Yuan[1], Honglei Zhuang[1], Tim Hanratty[2], and Jiawei Han[1]

[1]Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA

[2]U.S. Army Research Laboratory, Adelphi, MD, USA

[1]{czhang82, ll2, dlei5, qyuan, hzhuang3, hanj}@illinois.edu

[2]timothy.p.hanratty@mail.mil

ABSTRACT

Detecting local events (*e.g.*, protest, disaster) at their onsets is an important task for a wide spectrum of applications, ranging from disaster control to crime monitoring and place recommendation. Recent years have witnessed growing interest in leveraging geo-tagged tweet streams for online local event detection. Nevertheless, the accuracies of existing methods still remain unsatisfactory for building reliable local event detection systems. We propose TrioVecEvent, a method that leverages multimodal embeddings to achieve *accurate* online local event detection. The effectiveness of TrioVecEvent is underpinned by its two-step detection scheme. First, it ensures a high coverage of the underlying local events by dividing the tweets in the query window into coherent geo-topic clusters. To generate quality geo-topic clusters, we capture short-text semantics by learning multimodal embeddings of the location, time, and text, and then perform online clustering with a novel Bayesian mixture model. Second, TrioVecEvent considers the geo-topic clusters as candidate events and extracts a set of features for classifying the candidates. Leveraging the multimodal embeddings as background knowledge, we introduce discriminative features that can well characterize local events, which enable pinpointing true local events from the candidate pool with a small amount of training data. We have used crowdsourcing to evaluate TrioVecEvent, and found that it improves the performance of the state-of-the-art method by a large margin.

1 INTRODUCTION

Detecting local events (*e.g.*, disaster, protest, sport game) at their onsets is in pressing need for many applications. For example, in disaster control, it is highly important to build a real-time disaster detector that constantly monitors a geographical region. By sending out timely alarms when emergent disasters outbreak, the detector can help people take timely actions to alleviate huge life and economic losses. Another example is public order maintaining. For local governments, it is desirable to monitor people's activities in the city and know about social unrests (*e.g.*, protest, crime) as soon as possible. With a detector that discovers social unrests upon their onsets, the government can respond timely to prevent severe social riots.

While the task of online local event detection is extremely challenging years ago due to the lack of data sources, it has been recently made possible by the proliferation of geo-tagged social media. As a local event outbreaks, a considerable number of geo-tagged records (*e.g.*, tweets, Instagram posts) often emerge instantly [1, 10, 22, 24, 32, 44], created by the participants and/or witnesses who broadcast it right on the spot. Such records, as a result of human sensing, not only provide a comprehensive view of the event with multi-dimensional information (location, time, and text), but also serve as first-hand reports because of their real-time nature.

A handful of studies [1, 44] have investigated leveraging geo-tagged tweet streams for online local event detection. Typically, they cluster the keywords/tweets in the query window into candidate events, and then rank the candidates to select the top-K locally bursty ones. Despite the compelling results achieved by these studies, their detection accuracies remain unsatisfactory. For example, the state-of-the-art method [44] achieves only ∼30% precision, which is inadequate for building reliable local event detection systems in practice.

Indeed, online local event detection from continuous geo-tagged tweet streams is by no means a trivial task. It has several unique challenges that largely limit the performance of existing methods: 1) *Capturing short-text semantics.* To effectively extract local events, it is key to carefully capture the text semantics, such that the tweets discussing about the same event can be grouped. Existing methods, however, fall short in capturing short-text semantics. They either consider each keyword as an independent item [1], or rely on measures (*e.g.*, random walk [44]) that could suffer severely from text sparsity; 2) *Filtering uninteresting activities.* Many geo-tagged tweets just reflect routine activities (*e.g.*, dining, shopping) instead of any interesting events. To identify true local events, existing methods rely on heuristic ranking functions to select the top-K bursty candidates. However, it is hard — if not impossible — to manually design a gold-standard ranking function for accurate candidate filtering. Worse still, some query windows contain more than K events, while some (*e.g.*, the early morning) contain no events at all. The rigid top-K selection can incur severe detection accuracy loss due to its inflexibility; and 3) *Fast online detection.* When a local event outbreaks, our goal is to report the event instantly to allow for timely actions. Hence, it is desirable to continuously monitor the massive geo-tagged tweet stream and report local events on

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '17, August 13-17, 2017, Halifax, NS, Canada

© 2017 ACM. 978-1-4503-4887-4/17/08…$15.00

DOI: http://dx.doi.org/10.1145/3097983.3098027

the fly. Such a requirement renders existing batch-wise detection methods [8, 22, 36] inapplicable.

We propose TrioVecEvent, an embedding-based detector that enables *accurate* online local event detection from continuous geo-tagged tweet streams. The foundation of TrioVecEvent is a multimodal embedding learner that maps all the regions, hours, and keywords into the same space with their correlations preserved. If two units are highly correlated (*e.g.*, 'Pats' and 'Patriots', or the 5th Ave region and the keyword 'shopping'), their representations in the latent space tend be close. Such multimodal embeddings not only allow us to capture the subtle semantic similarities between tweets, but also serve as background knowledge by revealing the typical keywords in different regions and hours.

Built upon the multimodal embeddings, TrioVecEvent employs a two-step scheme to achieve high detection accuracy. First, *it performs online clustering to divide the tweets in the query window into coherent geo-topic clusters*. We develop a novel Bayesian mixture model that jointly models the tweet locations in the Euclidean space and the semantic embeddings in the spherical space. The model can generate quality candidates to ensure a high coverage of the underlying events. Second, *it extracts a set of discriminative features for accurate candidate classification.* Based on the multimodal embeddings, we design features that can well characterize local events, which enable pinpointing true positives from the candidate pool with a small amount of training data. Compared with existing top-K candidate selection schemes, the classification-based candidate filtering not only frees us from designing heuristic ranking functions, but also eliminates the inflexibility of rigid top-K selection. Furthermore, as the query window shifts continuously, TrioVecEvent does not need to detect the local events in the new window from scratch, but just needs to update the previous results with little cost to enable fast online detection.

Our main contributions are summarized as follows:

(1) We develop a multimodal embedding learner that jointly maps the location, time, and text into the same latent space with their correlations preserved. The multimodal embeddings not only capture the subtle semantics of tweet messages, but also serve as background knowledge to extract discriminative features for candidate filtering.
(2) We propose a novel Bayesian mixture clustering model that finds geo-topic clusters as candidate events. It generates quality geo-topic clusters without specifying the number of clusters a priori, and continuously updates the clustering results as the query window shifts.
(3) We design an effective candidate classifier that judges whether each candidate is indeed a local event. Relying on the multimodal embeddings, we extract a set of discriminative features for the candidates, which enable training a reliable classifier with a small amount of training data.

We have performed extensive experiments on two large-scale geo-tagged tweet data sets. Our effectiveness studies based on crowdsourcing show that TrioVecEvent improves the detection precision of the state-of-the-art method [44] by a large margin. Meanwhile, TrioVecEvent demonstrates excellent efficiency, making it suitable to be deployed for monitoring large geo-tagged tweet streams in practice.

2 RELATED WORK

In this section, we review existing work related to our problem, including: (1) event detection and forecasting; and (2) geo-tagged social media mining.

2.1 Event Detection and Forecasting

2.1.1 Global Event Detection. A larger number of methods have been proposed for extracting global events that are bursty in the entire data stream. Generally, existing global event detection approaches can be classified into two categories: *document-based* and *feature-based.* Document-based approaches [3, 4, 33] consider each document as a basic unit. They group similar documents into clusters and then find the bursty ones as events. For instance, Allan *et al.* [4] perform online clustering and use a similarity threshold to determine whether a new document should form a new topic or be merged into an existing one; Aggarwal *et al.* [3] also detect events via online clustering, but with a similarity measure that considers both tweet content relevance and user proximity; Sankaranarayanan *et al.* [33] train a Naïve Bayes filter to obtain news-related tweets and cluster them based on TF-IDF similarity. Feature-based approaches [17, 20, 23, 26, 37] identify a set of bursty features (*e.g.*, keywords) and cluster them to form events. Various techniques for extracting bursty features have been proposed, such as Fourier transform [17], Wavelet transform [37], and phrase-based burst detection [14, 23]. For example, Fung *et al.* [13] model feature occurrences with binomial distribution to extract bursty features; He *et al.* [17] construct the time series for each feature and perform Fourier Transform to identify bursts; Weng *et al.* [37] use wavelet transform and auto-correlation to measure word energy and extract high-energy words; Li *et al.* [23] segment each tweet into meaningful phrases and extract bursty phrases based on frequency; Giridhar *et al.* [14] extract an event as a group of tweets that contain at least one pair of bursty keywords.

The above methods are all designed for detecting globally bursty events. A local event, however, is usually bursty in a local region instead of the entire stream. Hence, directly applying these methods to our problem can miss many local events.

2.1.2 Local Event Detection. Local event detection has been receiving increasing research interest in the past few years [1, 8, 11, 12, 22, 30, 32, 44]. Watanabe *et al.* [36] and Quezada *et al.* [30] extract location-aware events in the social media, but their focus is on geo-locating the tweets/events. Sakaki *et al.* [32] achieve real-time earthquake detection, by training a classifier to judge whether an incoming tweet is earthquake-related. Li *et al.* [24] detect crime and disaster events (CDE) with a self-adaptive crawler for CDE-related tweets. Our work differs from these studies in that we aim to detect all kinds of local events, whereas they focus on specific event types. Quite a few generic local event detection methods have been proposed [1, 8, 22, 44]. Chen *et al.* [8] use Wavelet transform to extract spatiotemporally bursty Flickr tags, and then cluster them based on their co-occurrences and spatiotemporal distributions. Krumm *et al.* [22] discretize the time into equal-size bins and compare the number of tweets in the same bin across different days to extract local events.

Nevertheless, the above methods can only handle static data and detect local events in batch. While online methods have been gaining increasing attention in the data mining community [7, 18, 25, 39], few methods exist for supporting online local event detection. To the best of our knowledge, there are only two studies that have investigated the online local event detection problem [1, 44]. Abdelhaq *et al.* [1] first extract bursty and localized keywords in the query window, then cluster such keywords based on their spatial distributions, and finally select the top-K locally bursty clusters. Zhang *et al.* [44] detect geo-topic clusters based on random walk, and later rank these clusters to select spatiotemporally bursty ones. While these two methods support online local event detection, their accuracies are limited because of two reasons: 1) the clustering step does not capture short-text semantics well; and 2) the candidate filtering effectiveness is limited by heuristic ranking functions and the inflexibility of top-K selection.

2.1.3 Local Event Forecasting. Local event forecasting is another line of research that is related to our problem. Foley *et al.* [12] use distant supervision to extract future local events from Web pages, but the proposed method can only extract local events that are well advertised in advance on the Web. Zhao *et al.* [45–47] formulate local event forecasting as a binary prediction problem, *i.e.*, predicting whether a specific type of event (*e.g.*, civil unrest) will occur on a given day. Their methods combine social media with other data sources (*e.g.*, gold standard report, news articles) to train reliable predictors. Our problem is orthogonal to their studies in that, instead of performing binary prediction for a specific event type, we attempt to extract all types of local events at their onsets.

2.2 Geo-Tagged Social Media Mining

The emergence of geo-tagged social media has enabled progresses in various location-based mining tasks besides local event detection, including location recommendation [35], link prediction [9], mobility modeling [43], and spatiotemporal activity modeling [42]. Among such tasks, spatiotemporal activity modeling is mostly related to the local event detection problem, thus we detail existing approaches to it in the following.

Spatiotemporal activity modeling aims at detecting the typical activities in different geographical regions [6, 16, 21, 34]. Early approaches incorporate spatiotemporal information into classic topic models for this problem. In particular, Sizov *et al.* [34] extend LDA [6] by assuming each latent topic has a multinomial distribution over text and two Gaussians over latitudes and longitudes; Yin *et al.* [40] extend PLSA by assuming each region has a Gaussian distribution that generates locations, as well as a multinomial distribution over the latent topics that generate text; Guo *et al.* [16] use Dirichlet Process to extract activities that freely span several regions and peak multiple time periods; Hong *et al.* [19] and Yuan *et al.* [41] introduce the user factor in the modeling process to capture user preferences. Recently, representation learning has attracted much attention for mobile data mining tasks [38, 42, 48]. Among them, Zhang *et al.* [42] propose a multimodal representation learning method for the spatiotemporal activity modeling problem. By embedding geographical regions, time periods, and keywords into the same latent space, the proposed approach is capable of capturing their cross-modal correlations and outperforming previous geographical topic models. Although spatiotemporal activity modeling and local event detection are two closely related tasks, there is a clear difference between them. The former attempts to summarize the typical activities in different regions, whereas the latter aims at detecting unusual activities bursted in local areas.

3 PRELIMINARIES

3.1 Problem Description

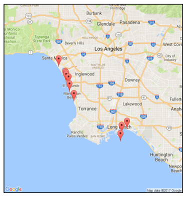
Let $\mathcal{D} = (d_1, d_2, \ldots, d_n, \ldots)$ be a continuous stream of geo-tagged tweets that arrive in chronological order. Each tweet d is a tuple $\langle t_d, l_d, x_d \rangle$, where t_d is its post time, l_d is its geo-location, and x_d is a bag of keywords that denote the tweet message. Consider a query time window $Q = [t_s, t_e]$ where t_s and t_e are the start and end timestamps satisfying $t_{d_1} \leq t_s < t_e \leq t_{d_n}$. The online local event detection problem aims at extracting all the local events that occur during Q and updating the event list online as Q shifts continuously.

3.2 The Framework of TrioVecEvent

A local event often results in relevant tweets around its occurring location. For example, suppose a protest occurs at the JFK Airport in New York City, many participants post tweets on the spot to express their attitude, with keywords like 'protest' and 'rights'. Such tweets form a geo-topic cluster as they are geographically close and semantically relevant. However, not necessarily does every geo-topic cluster correspond to a local event. It is because a geo-topic cluster may correspond to just routine activities in the region, *e.g.*, taking flights at JFK, shopping at the 5th Ave, *etc.*. We claim that a local event often leads to a *bursty and unusual geo-topic cluster*. The cluster is bursty in that it consists of a considerable number of tweets, and unusual in that its semantics deviates from routine activities significantly.

Motivated by the above, we design an embedding-based detection method TrioVecEvent. At the foundation of TrioVecEvent is a multimodal embedding learner [42] that maps all the regions, hours, and keywords into a latent space. If two units are highly correlated (*e.g.*, 'flight' and 'airport', or the JFK Airport region and the keyword 'flight'), their embeddings in the latent space tend be close. Figure 1 shows two real examples in Los Angeles and New York City, where we learn multimodal embeddings using millions of geo-tagged tweets in these cities and perform similarity searches. One can see that given the example queries, the multimodal embeddings well capture the correlations between different units. The usage of such embeddings is two-fold: 1) they allow us to capture the semantic similarities between tweets and further group the tweets into coherent geo-topic clusters; and 2) they reveal the typical keywords appearing in different regions and hours, which serve as background knowledge to help identify unusual and bursty geo-topic clusters.

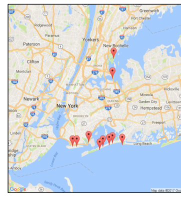
Figure 2 shows the framework of TrioVecEvent. As shown, *the embedding learner* embeds the location, time, and text using massive data from the geo-tagged tweet stream. It maintains a cache for keeping newly arrived tweets and updating the embeddings periodically. Based on the multimodal embeddings, TrioVecEvent employs a two-step detection scheme: 1) in *the online clustering step*, we develop a Bayesian mixture model that jointly models geographical locations and semantic embeddings to extract coherent geo-topic clusters in the query window; 2) in *the candidate classification* step, we extract a set of discriminative features for the



“beach”

“33.942, -118.409”	“nba”	“baseball”	“beach”
lax	lakers	dodgers	beachlife
international	kobe	ladders	sand
losangeles	bryant	dogerstadium	boardwalk
united	bulls	itfdb	ocean
people	cavs	letsgododgers	wave
tsa	kevin	game	beachday
sfo	knicks	dodgergame	pacificocean
food	clipper	play	santamonica
flight	lebron	losdoyers	pier
travel	cp3	win	wave

(a) Examples on LA (the second query is the location of the LAX Airport).



“beach”

“40.641, -73.778”	“nba”	“baseball”	“beach”
jfk	knicks	mlb	rockaway
airport	melo	yankees	beachday
international	lebron	mets	howard_beach
johnfkennedy	durant	yanks	brighton
burger	basketball	inning	longbeach
terminal	kobe	yankee	coney
john	cavs	ballpark	atlantic
kennedy	theknicks	pitch	island
sfo	game	jeter	boardwalk
flight	lakers	game	long

(b) Examples on NY (the second query is the location of the JFK Airport).

Figure 1: Example similarity queries based on the multimodal embeddings learned from the geo-tagged tweets in Los Angeles and New York City. In each city, the first query retrieves regions relevant to the keyword ‘beach’; the second retrieves keywords relevant to the airport location; and the last three retrieve relevant keywords for the given query keywords. For each query, we use the learned embeddings to compute the cosine similarities between different units, and retrieve the top ten most similar units without including the query itself.

candidates and determine whether each candidate is a true local event.

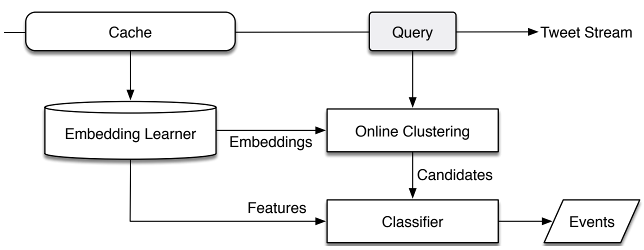


Figure 2: The framework of TrioVecEvent.

Now the key questions about TrioVecEvent are: 1) how to generate embeddings that can well capture the correlations between different units? 2) how do we perform online clustering to obtain quality geo-topic clusters in Q? and 3) what are the features that can discriminate true local events from non-events? In what follows, we introduce the multimodal embedding learner in Section 4, and then describe the two-step detection process of TrioVecEvent in Section 5.

4 MULTIMODAL EMBEDDING

The multimodal embedding module jointly maps all the spatial, temporal, and textual units into the same low-dimensional space with their correlations preserved. While the keywords are natural textual units for embedding, the space and time are are continuous and there are no natural embedding units. To address this issue, we break the geographical space into equal-size regions (300m*300m) and consider each region as a spatial unit. Similarly, we break one day into 24 hours and consider every hour as a basic temporal unit.

The multimodal embedding learner consumes the continuous tweet stream and learns D-dimensional representations for all the regions, hours, and keywords. As aforementioned, we maintain a cache C for keeping newly arrived tweets, and use it to periodically update the embeddings. To effectively incorporate the information in C without overfitting, we take the embeddings learned before the arrival of C as initialization, and optimize the embeddings over C for one full epoch. Such a simple strategy efficiently incorporates the tweets in the cache C, while largely preserving the information in the historical stream.

Our embedding procedure is inspired by the CBOW model [27] that predicts one unit given its context. Specifically, given a tweet d, for any unit $i \in d$ with type X (region, hour, or keyword), let $\mathbf{v}_i$ be the embedding of unit i, then we model the likelihood of observing i as

$$p(i|d_{-i}) = \exp(s(i, d_{-i})) / \sum_{j \in X} \exp(s(j, d_{-i})),$$

where d_{-i} is the set of all the units in d except i; and $s(i, d_{-i})$ is the similarity score between i and d_{-i}, defined as

$$s(i, d_{-i}) = \mathbf{v}_i^{\mathrm{T}} \sum_{j \in d_{-i}} \mathbf{v}_j / |d_{-i}|.$$

For a cache C of geo-tagged tweets, the objective is to predict all the units of the tweets in C:

$$J_C = -\sum_{d \in C} \sum_{i \in d} \log p(i|d_{-i}).$$

To efficiently optimize the above objective function, we follow the idea of negative sampling [27] and use stochastic gradient descent (SGD) for updating. At each time, we randomly sample a tweet d from C and a unit $i \in d$. With negative sampling, we randomly select K negative units that have the same type with i but do not appear in d. Then we minimize the following function for the selected samples:

$$J_d = -\log \sigma(s(i, d_{-i})) - \sum_{k=1}^{K} \log \sigma(-s(k, d_{-i})),$$

where $\sigma(\cdot)$ is the sigmoid function. The updating rules for different variables can be easily derived by taking the derivatives of the above objective and then applying SGD, we omit the details here due to the space limit.

5 TWO-STEP LOCAL EVENT DETECTION

In this section, we describe the two-step detection process of TrioVecEvent. We first develop a Bayesian mixture clustering model in Section 5.1, and then design a candidate classifier in Section 5.2. For clarity, Table 1 summarizes the notations used in this section.

Table 1: **The notations used in Section 5.**

$\mathcal{X}$	the set of semantic embeddings for the tweets in Q
$\mathcal{Z}$	the set of cluster memberships for the tweets in Q
$\mathcal{L}$	the set of geo-location vectors for the tweets in Q
$\boldsymbol{\kappa}$	the set of κ for all the clusters
$\boldsymbol{\kappa}^{\neg k}$	the subset of $\boldsymbol{\kappa}$ excluding the one for cluster k
$\mathbf{A}^{\neg d}$	the subset of any set $\mathbf{A}$ excluding element d
$\mathbf{A}^{k}$	the subset of elements that are assigned to cluster k in set $\mathbf{A}$
$\mathbf{x}^{k}$	the sum of the semantic embeddings in cluster k
$\mathbf{x}^{k,\neg d}$	the sum of the semantic embeddings in cluster k excluding d
n^{k}	the number of tweets in cluster k
$n^{k,\neg d}$	the number of tweets in cluster k excluding d

5.1 Candidate Generation

5.1.1 A Bayesian Mixture Clustering Model. We develop a Bayesian mixture clustering model to divide the tweets in the query window Q into a number of geo-topic clusters, such that the tweets in the same cluster are geographically close and semantically relevant. We consider each tweet d as a tuple $(\mathbf{l}_d, \mathbf{x}_d)$. Here, $\mathbf{l}_d$ is a 2-dimensional vector denoting d's geo-location; and $\mathbf{x}_d$ is the D-dimensional semantic embedding of d, derived by averaging the embeddings of the keywords in d's message.

The key idea behind our model is that every geo-topic cluster implies a coherent activity (*e.g.*, protest) around a certain geo-location (*e.g.*, the JFK Airport). The location acts as a *geographical center* that triggers geo-location observations around it in the Euclidean space; while the activity serves as a *semantic focus* that triggers semantic embedding observations around it in the spherical space. We assume there are at most K geo-topic clusters in the query window Q. Note that assuming the maximum number of clusters is a weak assumption that can be readily met in practice. At the end of the clustering process, some of these K cluster may become empty. As such, the appropriate number of clusters in any ad-hoc query window can be automatically discovered.

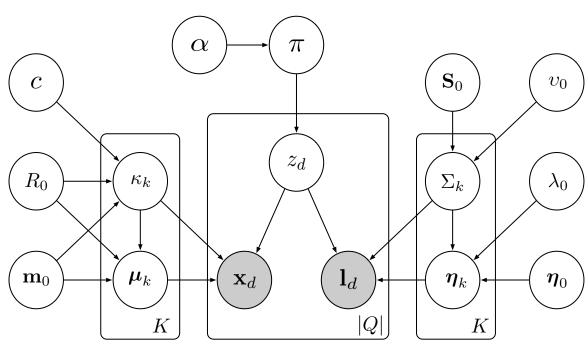


Figure 3: The graphical model of geo-topic clustering.

Figure 3 shows the generative process for all the tweets in the query window Q. As shown, we first draw a multinomial distribution π from a Dirichlet prior $\mathbf{Dirichlet}(.|\alpha)$ [28]. Meanwhile, for modeling the geo-locations, we draw K normal distributions from a Normal-Inverse-Wishart (NIW) prior $\mathbf{NIW}(.|\boldsymbol{\eta}_0, \lambda_0, \mathrm{S}_0, v_0)$ [28], which is a conjugate prior of the normal distribution; and for modeling the semantic embeddings, we draw K von Mises-Fisher (vMF) distributions from its conjugate prior $\Phi(\boldsymbol{\mu}, \kappa|\mathbf{m}_0, R_0, c)$ [29]. For each tweet $d \in Q$, we first draw its cluster membership z_d from π. Once the cluster membership is determined, we draw its geo-location $\mathbf{l}_d$ from the respective normal distribution, and its semantic embedding $\mathbf{x}_d$ from the respective vMF distribution.

While using normal distributions for modeling the geo-location $\mathbf{l}_d$ is intuitive, we justify the choice of the vMF distribution for modeling the semantic embedding $\mathbf{x}_d$ as follows. For a D-dimensional unit vector $\mathbf{x}$ that follows vMF distribution, its probability density function is given by

$$p(\mathbf{x}|\boldsymbol{\mu}, \kappa) = C_D(\kappa)\exp(\kappa\boldsymbol{\mu}^T\mathbf{x}),$$

where $C_D(\kappa) = \frac{\kappa^{D/2-1}}{I_{D/2-1}(\kappa)}$ and $I_{D/2-1}(\kappa)$ is the modified Bessel function. The vMF distribution has two parameters: the mean direction $\boldsymbol{\mu}$ ($\boldsymbol{\mu} = \mathbf{1}$) and the concentration parameter κ ($\kappa > 0$). The distribution of $\mathbf{x}$ on the unit sphere concentrates around the mean direction $\boldsymbol{\mu}$, and is more concentrated if κ is large. Our choice of the vMF distribution is motivated by the effectiveness of the cosine similarity [27, 42] in quantifying the similarities between multimodal embeddings. The mean direction $\boldsymbol{\mu}$ acts as a semantic focus on the unit sphere, and produces relevant semantic embeddings around it, where concentration degree is controlled by the parameter κ. The superiority of the vMF distribution over other alternatives (*e.g.*, Gaussian) for modeling textual embeddings has also been demonstrated in recent studies on clustering [15] and topic modeling [5].

To summarize the above generative process, we have:

$$\begin{aligned}
&\pi \sim \mathbf{Dirichlet}(.|\alpha)\\
&\{\boldsymbol{\eta}_k, \Sigma_k\} \sim \mathbf{NIW}(.|\boldsymbol{\eta}_0, \lambda_0, \mathrm{S}_0, v_0) \quad k = 1, 2, \ldots, K\\
&\{\boldsymbol{\mu}_k, \kappa_k\} \sim \Phi(.|\mathbf{m}_0, R_0, c) \quad k = 1, 2, \ldots, K\\
&z_d \sim \mathbf{Categorical}(.|\pi) \quad d \in Q\\
&\mathbf{l}_d \sim \mathcal{N}(.|\boldsymbol{\eta}_{z_d}, \Sigma_{z_d}) \quad d \in Q\\
&\mathbf{x}_d \sim \mathbf{vMF}(.|\boldsymbol{\mu}_{z_d}, \kappa_{z_d}) \quad d \in Q
\end{aligned}$$

where $\Lambda = \{\alpha, \mathbf{m}_0, R_0, c, \boldsymbol{\eta}_0, \lambda_0, \mathrm{S}_0, v_0\}$ are the hyper-parameters for the prior distributions.

5.1.2 Parameter Estimation. The key to obtain the geo-topic clusters is to estimate the posterior distributions for $\{z_d\}_{d\in Q}$. We use Gibbs sampling for this purpose. Since we have chosen conjugate priors for π and $\{\boldsymbol{\mu}_k, \boldsymbol{\eta}_k, \Sigma_k\}_{k=1}^K$, these parameters can be integrated out during the Gibbs sampling process, resulting in a collapsed Gibbs sampling procedure. Due to the space limit, we directly give the conditional probabilities for $\{\kappa_k\}_{k=1}^K$ and $\{z_d\}_{d\in Q}$:

$$p(\kappa_k|\boldsymbol{\kappa}^{\neg k}, \mathcal{X}, \mathcal{Z}, \alpha, \mathbf{m}_0, \mathcal{R}_0, c) \propto \frac{(C_D(\kappa_k))^{c+n^k}}{C_D(\kappa_k\|R_0\mathbf{m}_0 + \mathbf{x}^k\|)}, \tag{1}$$

$$\begin{aligned}
&p(z_d = k|\mathcal{X}, \mathcal{L}, \mathcal{Z}^{\neg d}, \boldsymbol{\kappa}, \Lambda) \propto p(z_d = k|\mathcal{Z}^{\neg d}, \alpha)\cdot\\
&p(\mathbf{x}_d|\mathcal{X}^{\neg d}, \mathcal{Z}^{\neg d}, z_d = k, \Lambda)\cdot p(\mathbf{l}_d|\mathcal{L}^{\neg d}, \mathcal{Z}^{\neg d}, z_d = k, \Lambda).
\end{aligned} \tag{2}$$

The three quantities in Equation 2 are given by:

$$p(z_d = k|\cdot) \propto (n^{k,\neg d} + \alpha), \tag{3}$$

$$p(\mathbf{x}_d|\cdot) \propto \frac{C_D(\kappa_k) C_D(\|\kappa_k(R_0\mathbf{m}_0 + \mathbf{x}^{k,\neg d})\|_2)}{C_D(\|\kappa_k(R_0\mathbf{m}_0 + \mathbf{x}^{k,\neg d} + \mathbf{x}_d)\|_2)}, \tag{4}$$

$$p(\mathbf{l}_d|\cdot) \propto \frac{\lambda^{k,\neg d}(\upsilon^{k,\neg d} - 1)|\mathbf{S}^{\mathcal{L}^k \cap \mathcal{L}^{\neg d}}|^{\upsilon^{k,\neg d}/2}}{2(\lambda^{k,\neg d} + 1)|\mathbf{S}^{\mathcal{L}^k \cup \{\mathbf{l}_d\}}|^{(\upsilon^{k,\neg d}+1)/2}}, \tag{5}$$

where $\lambda^{\cdot}$, $\upsilon^{\cdot}$, and $\mathbf{S}^{\cdot}$ are the posterior estimations for the parameters of the NIW distribution [28].

From Equation 2, 3, 4, and 5, we observe that our Bayesian mixture model enjoys several nice properties when determining the cluster membership for a tweet d: 1) With Equation 3, d tends to join a cluster that has more members, resulting in a rich-get-richer effect; 2) With Equation 4, d tends to join a cluster that is more semantically similar to its textual embedding $\mathbf{x}_d$, leading to semantically coherent clusters; and 3) With Equation 5, d tends to join a cluster that is more geographically close to its geo-location $\mathbf{l}_d$, resulting in geographically compact clusters.

5.1.3 Incremental Updating. When the query window Q shifts, it is undesirable to re-compute the geo-topic clusters in the new query window from scratch for the purpose of fast online detection. We employ an incremental updating strategy that efficiently approximates the clustering results in the new window. As shown in Figure 4, assume the query window shifts from Q to Q', we denote by $D_- = \{d_1, \ldots, d_m\}$ the outdated tweets, and $D_+ = \{d_{n-k+1}, \ldots, d_n\}$ the new tweets. Instead of performing Gibbs sampling for all the tweets in Q', we simply drop D_- and sample the cluster memberships for the tweets in D_+. Such an incremental updating strategy achieves excellent efficiency and yields quality geo-topic clusters in practice as the memberships of the remaining tweets are mostly stable.

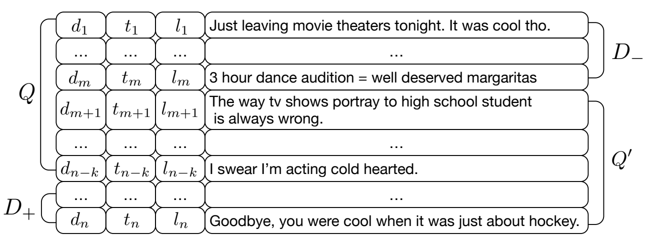


Figure 4: Incremental updating as the query window shifts.

5.2 Candidate Classification

We have so far obtained a set of coherent geo-topic clusters in the query window as candidates. Now we proceed to describe the candidate classifier for pinpointing the true local events. The key is to define a set of features that can well discriminate true local events from non-events. We introduce the following set of features for this purpose:

(1) **Spatial unusualness** quantifies how unusual a candidate is in its geographical region. As the multimodal embeddings can unveil the typical keywords in different regions, we use them as background knowledge to measure the spatial unusualness of a candidate C. Specifically, we compute the spatial unusualness as $f_{su}(C) = \sum_{d \in C} \cos(\mathbf{v}_{l_d}, \mathbf{x}_d)/|C|$, where $\mathbf{v}_{l_d}$ is the embedding of the region of tweet d, and $\mathbf{x}_d$ is the semantic embedding of tweet d.

(2) **Temporal unusualness** quantifies how temporally unusual a candidate is. We define the temporal unusualness of a candidate C as $f_{tu}(C) = \sum_{d \in C} \cos(\mathbf{v}_{t_d}, \mathbf{x}_d)/|C|$, where $\mathbf{v}_{t_d}$ is the embedding of the hour of tweet d.

(3) **Spatiotemporal unusualness** jointly considers the space and time to quantify how unusual a candidate C is: $f_{stu}(C) = \sum_{d \in C} \cos((\mathbf{v}_{l_d} + \mathbf{v}_{t_d})/2, \mathbf{x}_d)/|C|$.

(4) **Semantic concentration** computes how semantically coherent C is: $f_{su}(C) = \sum_{d \in C} \cos(\overline{\mathbf{x}}_d, \mathbf{x}_d)/|C|$, where $\overline{\mathbf{x}}_d$ is the average semantic embedding of the tweets in C.

(5) **Spatial and temporal concentrations** quantify how concentrated a candidate C is over the space and time. We compute three quantities for the tweets in C: 1) the standard deviation of the longitudes; 2) the standard deviation of the latitudes; and 3) the standard deviation of the creating timestamps.

(6) **Burstiness** quantifies how bursty a candidate C is. We define it as the number of tweets in C divided by the time span of C.

The Classification Procedure. To summarize, for each candidate C, we extract the following features: 1) the spatial unusualness; 2) the temporal unusualness; 3) the spatiotemporal unusualness; 4) the semantic concentration; 5) the longitude concentration; 6) the latitude concentration; 7) the temporal concentration; and 8) the burstiness. With the above features, we use logistic regression to train a binary classifier and judge whether each candidate is indeed a local event. We choose the logistic regression classifier because of its robustness when there is limited training data. We have also tried other classifiers like Random Forest, and find that the logistic regression classifier has slightly better performance in our experiments. The training instances are collected over 100 query windows in a crowdsourcing platform. We will shortly describe the labeling process in Section 6.

5.3 Complexity Analysis

We analyze the time complexities of the candidate generation step and the candidate classification step separately. For candidate generation, to extract geo-topic clusters in the new query window, the time complexity is $O(INKD)$, where I is the number of Gibbs sampling iterations, N is the number of new tweets, K is the maximum number of clusters; and D is the latent embedding dimension. Note that I, K and D are usually fixed to moderate values in practice, thus the candidate generation step scales roughly linearly with N and has good efficiency. For candidate classification, the major overhead lies in feature extraction. Let N_c be the maximum number of tweets in each candidate, then the time complexity of feature extraction is $O(KN_cD)$.

6 EXPERIMENTS

6.1 Experimental Settings

6.1.1 Baselines. We compare TrioVecEvent with all the existing online local event detection methods that we are aware of, described as follows:

- EvenTweet [1] extracts bursty and localized keywords from the query window, then clusters these keywords based on spatial distributions, and finally selects top-K locally bursty clusters.
- GeoBurst [44] is the state-of-the-art method for online local event detection. It first uses random walk on a keyword co-occurrence graph to detect geo-topic clusters, and then ranks all the clusters by the weighted combination of spatial burstiness and temporal burstiness.
- GeoBurst+ is an upgraded version of GeoBurst by replacing the ranking module with a classifier. Instead of heuristically ranking the candidates, we train a classifier to determine whether each candidate is a local event. The used features include spatial burstiness, temporal burstiness [44], as well as spatial and temporal concentrations (Section 5.2).

6.1.2 Parameters. As EvenTweet and GeoBurst both perform top-K selection to identify local events from the candidate pool, we set $K = 5$ for them to achieve a tradeoff between precision and recall. Meanwhile, EvenTweet requires to partition the whole space into $M \times M$ small grids. After tuning, we set $M = 50$. In GeoBurst and GeoBurst+, there are three additional parameters: the kernel bandwidth h; (2) the restart probability α; and (3) the RWR similarity threshold δ. Following the original paper [44], we set them as $h = 0.01, \alpha = 0.2$, and $\delta = 0.02$. All the baseline methods require a reference window that precedes the query to quantify the burstiness of the candidates, we follow [44] and set the reference duration to one week.

TrioVecEvent involves the following major parameters: (1) the latent dimension D for embedding; and (2) the maximum number of clusters K; and (3) the number of Gibbs sampling iterations I. After tuning, we set $D = 100$, $K = 500$, and $I = 10$, as we find such a setting can produce geo-topic clusters that are fine-grained enough while achieving good efficiency. In addition, the Bayesian mixture model involves several hyper-parameters, as shown in Figure 3. In general, we observe that our model is not very sensitive to them. We set $\alpha = 1.0, c = 0.01, R_0 = 0.01, \mathbf{m}_0 = 0.1 \cdot \mathbf{1}, \lambda_0 = 1.0, \boldsymbol{\eta}_0 = \mathbf{0}, \upsilon_0 = 2.0, \mathbf{S}_0 = 0.01 \cdot \mathbf{I}$, which are commonly adopted values for the prior distributions used in our model. We conduct the experiments on a computer with Intel Core i7 2.4GHz CPU and 8GB memory.

6.1.3 Data Sets and Groundtruth. Our experiments are based on the same data sets as in [44]. The first data set LA consists of the geo-tagged tweets in Los Angeles collected during 2014.08.01 — 2014.11.30; and the second data set NY consists of the geo-tagged tweets in New York City during the same period. For each data set, we use an off-the-shelf tool [31] to preprocess the text messages by preserving entities and nouns, and then remove the keywords that appear less than 100 times in the entire corpus.

To evaluate the methods and collect training data for GeoBurst+ and TrioVecEvent, we randomly generate 200 non-overlapping query windows with four different lengths: 3-hour, 4-hour, 5-hour, and 6-hour. After ranking these windows in chronological order, we run each the method online by shifting a fixed-length (3h, 4h, 5h, 6h) query window on a 5-minute basis, and save the results falling in each target query window. After collecting labeled data with crowdsourcing, we use the groundtruth in the first 100 windows for training the classifiers of GeoBurst+ and TrioVecEvent; and that in the rest 100 windows for comparing all the methods.

Now we describe the labeling process based on crowdsourcing. For all the methods, we upload their results to CrowdFlower[1] for human judging. Since EvenTweet and GeoBurst are top-K methods with $K = 5$, we upload five results for each of them in each query window. GeoBurst+ and TrioVecEvent are classification-based methods, and the raw numbers of candidate events could be large. To limit the number of candidates while ensuring the coverages of the two methods, we employ a simple heuristic for eliminating negative candidates. It removes the candidates that have too few users (*i.e.*, the number of users is less than five) or too dispersed spatial distributions (*i.e.*, the longitude or latitude standard deviation is larger than 0.02). After filtering such trivial negatives, we upload the remaining candidates for evaluation.

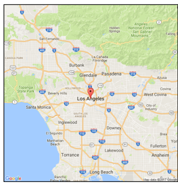
On CrowdFlower, we represent each event with five tweets and ten keywords, and ask three CrowdFlower workers to judge whether the event is indeed a local event. To ensure the quality of the workers, we label 20 queries as groundtruth judgments on each data set, such that only the workers who can achieve no less than 80% accuracy on the groundtruth can submit their answers. Finally, we use majority voting to aggregate the workers' answers. The representative tweets and keywords are selected as follows: (1) For GeoBurst and GeoBurst+, we select five tweets having the largest authority scores, and ten keywords having the largest TF-IDF weights. (2) EvenTweet represents each event as a group of keywords. We select ten keywords with the highest scores in each event. Then we regard the group of keywords as a query to retrieve the top five most similar tweets using the BM25 retrieval model. (3) TrioVecEvent represents a candidate as a group of tweets. We first compute the average semantic embedding, and then select the closest keywords and tweets using cosine similarity.

6.1.4 Metrics. As aforementioned, we use the groundtruth in the last 100 query windows to evaluate all the methods. To quantify the performance of all the methods, we report the following metrics: (1) *Precision.* The detection precision is $P = N_{\text{true}}/N_{\text{report}}$, where N_{true} is the number of true local events and N_{report} is the total number of reported events. (2) *Pseudo Recall.* The true recall is hard to measure due to the lack of the comprehensive set of events in the physical world. We thus measure the pseudo recall for each method. Specifically, for each query window, we aggregate the true positives of different methods. Let N_{total} be the total number of distinct local events detected by all the methods; we compute the pseudo recall of each method as $R = N_{\text{true}}/N_{\text{total}}$. (3) *Pseudo F1-Score.* Finally, we also report the pseudo F1 score of each method, which is computed as $F1 = 2 * P * R/(P + R)$.

6.2 Illustrative Cases

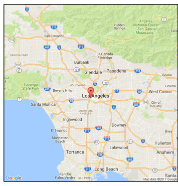
Before reporting the quantitative results, we first present several examples for TrioVecEvent. Figure 5 and 6 show several exemplifying geo-topic clusters detected by TrioVecEvent on LA and NY, respectively. For each cluster, we plot the locations of the member tweets and show the top five representative tweets. The clusters in Figure 5(a) and 5(b) correspond to two positive local events in LA: 1) a protesting rally held at the LAPD Headquarter for making voice for Mike Brown and Ezell Ford; and 2) Katy Perry's concert at the Staples Center. For each event, one can see the generated geo-topic

[1] http://www.crowdflower.com/



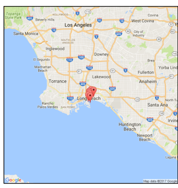
- Standing for **justice**! @ LAPD Headquarters http://t.co/YxNUAloQcE
- At the LAPD **protest** downtown **#EzellFord #MikeBrown** http://t.co/kWphv6dXOr
- Hands Up. Don't **Shoot**. @ Los Angeles City Hall
- Black, Brown, poor white, ALL **oppressed** people **unite**. #ftp #lapd **#ferguson** #lapd **#mikebrown #ezellford** http://t.co/szf3mJRJwV
- Finished **marching** now **gathered** back at LAPD police as organizers speak some truth **#EzellFord #MikeBrown #ferguson** http://t.co/M33n9IMOzC

(a) LA local event I: a protest rally at the LAPD Headquarter.



- Thanks for making my Teenage Dreams come true @arjanwrites!! AHHH @**KATYPERRY**!! (at @**STAPLESCenter** for **Katy Perry**) https://t.co/TVEaghr1Tt
- **Katy perry** with my favorite. http://t.co/FpfPYAQNBR
- @MahoganyLOX are you at the **Katy perry concert**?
- One of the beeeeest **concerts** in history!
- My two minutes of fame was me and my friends picture getting put on the TV screens at the **Katy Perry concert**.

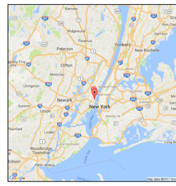
(b) LA local event II: Katy Perry's concert at the Staples Center.



- #beachlife @ Long Beach Shoreline Marina
- Downtown LB at night #DTLB #LBC #Harbor @ The Reef Restaurant
- Jambalaya @ California Pizza Kitchen at Rainbow Harbor http://t.co/9XbDhQAVsN
- #coachtoldmeto @ Octopus Long Beach http://t.co/lYQc8u2m1F
- El Sauz tacos are the GOAT.

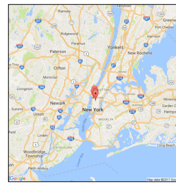
(c) LA non-event: enjoying beach life at the Long Beach.

Figure 5: Example geo-topic clusters on LA. The first two are classified as positive local events and the third as negative.



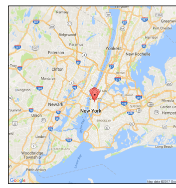
- **Hoboken Fall Arts & Music Festival** with bae @alli_holmes93 @ Washington St. Hoboken
- On Washington Street. (at **Hoboken Music And Arts Festival**) https://t.co/YbLSdZhLZV
- Sweeeeet. Bonavita **Guitars**, at the **Hoboken festival**. http://t.co/2Cw1Qz4UGo
- I'm at **Hoboken Music And Arts Festival** in Hoboken, NJ https://t.co/i4bSM3mrjb
- It's a **festy music** day.

(a) NY local event I: the Hoboken Music and Arts Festival in Hoboken, NJ.



- **Knicks game** w literally a person. http://t.co/hxVYidpCzs
- **Knicks game** with my main man.
- It has been one of my dream to watch **NBA game**!! Let's go! http://t.co/GRJRvFw6vd
- Watching @nyknicks at @TheGarden for for the first time! Go **Knicks**! #nyk4troops
- I was outside of **msg** today pretending I liked the **Knicks**. It's that bad.

(b) NY local event II: The Knicks' basketball game at the Madison Square Garden.



- Happiness is a shroom burger from Shake Shack. @ Shake Shack Times Square http://t.co/tvYqYbsK0o
- Just A Taco in the City ya Know #TimeSquare#DallasBBQ @ Dallas BBQ http://t.co/hyCNkpSrSd
- Craving a lobster roll, aka I must get to RI NOW.
- Rainbow Set Sushi dulu dan menikmati midtown manhattan sebelum kembali ke??? (at Wasabi Sushi & Bento) https://t.co/uC9rt8yCoC
- Pork carnitas tacos & blood orange margaritas w my favorite rican @ Lucys Cantina

(c) NY non-event: having food aroun the Time Square.

Figure 6: Example geo-topic clusters detected on NY. The first two are classified as positive local events; while the third as negative.

cluster is of high quality — the tweets in each cluster are highly geographically compact and semantically coherent. Even if there are tweets discussing about the event with different keywords (*e.g.*, 'shoot', 'justice', and 'protest'), TrioVecEvent can group them into the same cluster. This is because the multimodal embeddings can effectively capture the subtle semantic correlations between the keywords. While the first two clusters are classified as true local events by TrioVecEvent, the last one in Figure 5(c) is marked as negative. Although the last one is also a meaningful geo-topic cluster, it reflects routine activities around the long beach instead of any unusual events. TrioVecEvent is able to capture this fact and classify it into the negative class.

Figure 6(a) and 6(b) show two example local events detected by TrioVecEvent on NY. The first is the Hoboken Arts and Music Festival; and the second is the basketball game between the Knicks and the Hawks. Again, we can see the member tweets are highly relevant both geographically and semantically. As they represent interesting and unusual activities in their respective areas, TrioVecEvent successfully classifies them as true local events. In contrast, the third cluster just reflects the everyday activity of having food around the Time Square, and is returned as a non-event.

To further understand why TrioVecEvent is capable of generating high-quality geo-topic clusters and eliminating non-event candidates, we can re-examine the cases in Figure 1. As shown, the retrieved results based on the learned embeddings are highly meaningful. For instance, given the query 'beach', the top locations are all beach-life areas in LA and NYC; given the location of the airport, the top keywords reflect typical flight-related activities around the airport; and given different keywords as queries, the retrieved keywords are semantically relevant. Such results explain why TrioVecEvent is capable of grouping relevant tweets into the same geo-topic cluster and why the embeddings can serve as useful knowledge for extracting discriminative features (*e.g.*, spatial and temporal unusualness, semantic concentration).

6.3 Quantitative Results

6.3.1 Effectiveness Comparison. Table 2 reports the precision, pseudo recall, and pseudo F1 of all the methods on LA and NY. We find that TrioVecEvent significantly outperforms the baseline methods on both data sets. Compared with the strongest baseline GeoBurst+, TrioVecEvent yields around 118% improvement in precision, 26% improvement in pseudo recall, and 66% improvement in pseudo F1-score. The huge improvements are attributed to the two advantages of TrioVecEvent: (1) the embedding-based clustering model capture short-text semantics more effectively, and generate high-quality geo-topic clusters to achieve a good coverage of all the potential events; and (2) the multimodal embeddings enable the classifier to extract discriminative features for the candidates, and thus accurately pinpoint true local events.

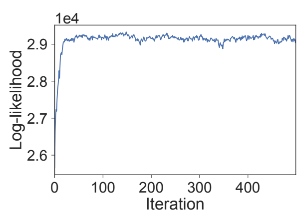
Comparing GeoBurst and its upgraded version GeoBurst+, we find that GeoBurst+ outperforms GeoBurst by a considerable margin. Such a phenomenon further verifies that classification-based candidate filtering is superior to the ranking-based strategy, even with moderately-sized training data. EvenTweet performs much poorer than the other methods on our data. After investigating the results, we find that although EvenTweet can extract

Table 2: The performance of different methods. 'P' is precision, 'R' is pseudo recall; and 'F1' is pseudo F1 score.

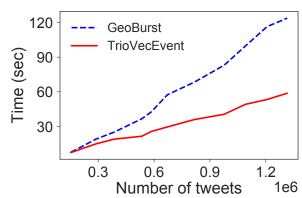
Method	LA			NY		
	P	R	F1	P	R	F1
EvenTweet	0.132	0.212	0.163	0.108	0.196	0.139
GeoBurst	0.282	0.451	0.347	0.212	0.384	0.273
GeoBurst+	0.368	0.483	0.418	0.351	0.465	0.401
TrioVecEvent	**0.804**	**0.612**	**0.695**	**0.765**	**0.602**	**0.674**

spatiotemporally bursty keywords in the query window, clustering these keywords merely based on the spatial distributions often leads to semantically irrelevant keywords in the same cluster, which yields suboptimal detection accuracies.

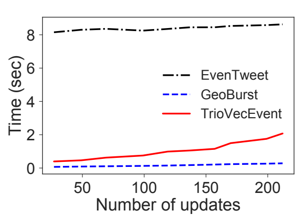
6.3.2 Efficiency Comparison. We proceed to report the efficiency of different methods. Since the time cost of GeoBurst+ is almost the same as GeoBurst, we only show the cost of GeoBurst for brevity. First, we study the convergence rate of the Gibbs sampler for the Bayesian mixture model. For this purpose, we randomly select a three-hour query window, and apply the Bayesian mixture model for extracting geo-topic clusters in the query window. Figure 7(a) shows the log-likelihood as the number of Gibbs sampling iterations increases. We observe that the log-likelihood quickly converges after a few iterations. Hence, it is usually sufficient to set the number of iterations to a relatively small value (*e.g.*, 10) in practice for better efficiency.



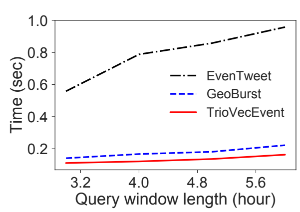
(a) Geo-topic clustering convergence.



(b) Summarization throughput.



(c) Online clustering time.



(d) Candidate filtering time.

Figure 7: Efficiency study on LA. Figure 7(a) shows the convergence rate of the Bayesian mixture model; Figure 7(b) shows the summarization throughputs for GeoBurst and TrioVecEvent; Figure 7(c) shows the cost of online clustering; and Figure 7(d) shows the cost of candidate filtering.

Both GeoBurst and TrioVecEvent require summarizing the continuous tweet stream for obtaining background knowledge: the summarization of GeoBurst is done by extending the Clustream algorithm [2]; while that of TrioVecEvent is achieved with multimodal embedding. In this set of experiments, we compare the throughputs of the summarization modules in these two methods. Specifically, we apply the two methods to process LA and record the accumulated CPU time for summarization in the process. As depicted in Figure 7(b), the summarization of both methods scales well with the number of tweets, and TrioVecEvent is about 50% faster than GeoBurst. Meanwhile, we observe that the embedding learner scales roughly linearly with the number of processed tweets, making it suitable for large-scale tweet streams.

Now we investigate the efficiency of online clustering and candidate filtering for different methods. To this end, we randomly generate 1000 3-hour query window, and continuously shift each query window on a basis of 1, 2, . . ., 10 minutes. In Figure 7(c), we report the averaged running time of different methods in terms of the number of new tweets. As shown, both GeoBurst and TrioVecEvent are much more efficient than EvenTweet, while GeoBurst is the fastest. In terms of candidate filtering, Figure 7(d) reports the running time of the three methods as the query window length changes. Among the three methods, TrioVecEvent achieves the best efficiency for candidate filtering. This is because TrioVecEvent needs to extract only a small set of features for candidate classification. With the learned multimodal embeddings, all of the features are quite cheap to compute.

6.3.3 Feature Importance. Finally, we measure the importance of different features for candidate classification. Our measurement is based on the Random Forest Classifier, by computing how many times a feature is used for dividing the training samples in the learned tree ensemble. Figure 8 plots the normalized fractions of all the features, where larger values indicate higher importance. As shown, the spatial concentrations turn out to be the most important features on both data sets. This is expected, as a local event usually occurs at a specific point-of-interest, resulting in a geo-topic cluster that is spatially compact. The unusualness measures also serve as important indicators for the classifier, which clearly shows that the embeddings serve as useful knowledge for distinguishing unusual events from routine activities. The other four features (burstiness, semantic concentration, spatiotemporal unusualness, and temporal concentration) act as useful indicators as well, receiving considerable weights for candidate classification.

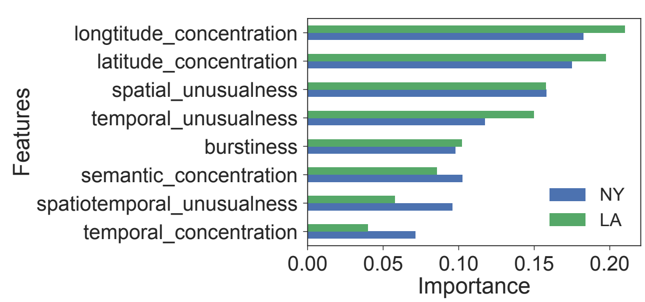


Figure 8: The importance of different features for candidate classification on LA and NY.

7 CONCLUSION

We have proposed the TrioVecEvent method to enable accurate online local event detection in continuous geo-tagged tweet streams. With the multimodal embeddings of the location, time, and text, TrioVecEvent first obtains quality geo-topic clusters in the query window to ensure a high coverage of the underlying events. It then extracts a set of features to characterize the candidates, such that the true local events can be accurately identified. Our extensive experiments have demonstrated that TrioVecEvent improves the accuracy of the state-of-the-art method significantly while achieving good efficiency. In the future, we are interested in extending the method for assigning true local events into fine-grained types, thus allowing the end users to take different actions for different event types.

8 ACKNOWLEDGEMENTS

We would like to thank the reviewers for their insightful comments. This work was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National Science Foundation IIS-1017362, IIS-1320617, and IIS-1354329, HDTRA1-10-1-0120, and Grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov), and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

[1] H. Abdelhaq, C. Sengstock, and M. Gertz. Eventweet: Online localized event detection from twitter. *PVLDB*, 6(12):1326–1329, 2013.
[2] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *VLDB*, pages 81–92, 2003.
[3] C. C. Aggarwal and K. Subbian. Event detection in social streams. In *SDM*, pages 624–635, 2012.
[4] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *SIGIR*, pages 37–45, 1998.
[5] K. Batmanghelich, A. Saeedi, K. Narasimhan, and S. Gershman. Nonparametric spherical topic modeling with word embeddings. In *ACL*, 2016.
[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(1):993–1022, 2003.
[7] L. Cao, M. Wei, D. Yang, and E. A. Rundensteiner. Online outlier exploration over large datasets. In *KDD*, pages 89–98, 2015.
[8] L. Chen and A. Roy. Event detection from flickr data through wavelet-based spatial analysis. In *CIKM*, pages 523–532, 2009.
[9] J. Cranshaw, E. Toch, J. I. Hong, A. Kittur, and N. M. Sadeh. Bridging the gap between physical location and online social networks. In *UbiComp*, pages 119–128, 2010.
[10] S. Doan, B.-K. H. Vo, and N. Collier. An analysis of twitter messages in the 2011 tohoku earthquake. In *Electronic Healthcare*, pages 58–66. Springer, 2012.
[11] W. Feng, C. Zhang, W. Zhang, J. Han, J. Wang, C. Aggarwal, and J. Huang. Streamcube: Hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream. In *ICDE*, pages 1561–1572, 2015.
[12] J. Foley, M. Bendersky, and V. Josifovski. Learning to extract local events from the web. In *SIGIR*, pages 423–432, 2015.
[13] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *VLDB*, pages 181–192, 2005.
[14] P. Giridhar, S. Wang, T. F. Abdelzaher, J. George, L. Kaplan, and R. Ganti. Joint localization of events and sources in social networks. In *DCOSS*, pages 179–188, 2015.
[15] S. Gopal and Y. Yang. Von mises-fisher clustering models. In *ICML*, pages 154–162, 2014.
[16] J. Guo and Z. Gong. A nonparametric model for event discovery in the geospatial-temporal space. In *CIKM*, pages 499–508, 2016.
[17] Q. He, K. Chang, and E.-P. Lim. Analyzing feature trajectories for event detection. In *SIGIR*, pages 207–214, 2007.
[18] X. He, H. Zhang, M. Kan, and T. Chua. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*, pages 549–558, 2016.
[19] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *WWW*, pages 769–778, 2012.
[20] W. Kang, A. K. H. Tung, W. Chen, X. Li, Q. Song, C. Zhang, F. Zhao, and X. Zhou. Trendspedia: An internet observatory for analyzing and visualizing the evolving web. In *ICDE*, pages 1206–1209, 2014.
[21] C. C. Kling, J. Kunegis, S. Sizov, and S. Staab. Detecting non-gaussian geographical topics in tagged photo collections. In *WSDM*, pages 603–612, 2014.
[22] J. Krumm and E. Horvitz. Eyewitness: Identifying local events via space-time signals in twitter feeds. In *SIGSPATIAL*, 2015.
[23] C. Li, A. Sun, and A. Datta. Twevent: segment-based event detection from tweets. In *CIKM*, pages 155–164, 2012.
[24] R. Li, K. H. Lei, R. Khadiwala, and K.-C. Chang. Tedas: A twitter-based event detection and analysis system. In *ICDE*, pages 1273–1276, 2012.
[25] S. Liang, E. Yilmaz, and E. Kanoulas. Dynamic clustering of streaming short documents. In *KDD*, pages 995–1004, 2016.
[26] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *SIGMOD*, pages 1155–1158, 2010.
[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
[28] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
[29] G. Nunez-Antonio and E. Gutiérrez-Pena. A bayesian analysis of directional data using the von mises–fisher distribution. *Communications in StatisticsfiSimulation and Computation®*, 34(4):989–999, 2005.
[30] M. Quezada, V. Peña-Araya, and B. Poblete. Location-aware model for news events in social media. In *SIGIR*, pages 935–938, 2015.
[31] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, pages 1524–1534, 2011.
[32] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, pages 851–860, 2010.
[33] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: news in tweets. In *GIS*, pages 42–51, 2009.
[34] S. Sizov. Geofolk: latent spatial semantics in web 2.0 social media. In *WSDM*, pages 281–290, 2010.
[35] W. Wang, H. Yin, L. Chen, Y. Sun, S. W. Sadiq, and X. Zhou. Geo-sage: A geographical sparse additive generative model for spatial item recommendation. In *KDD*, pages 1255–1264, 2015.
[36] K. Watanabe, M. Ochi, M. Okabe, and R. Onai. Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs. In *CIKM*, pages 2541–2544, 2011.
[37] J. Weng and B.-S. Lee. Event detection in twitter. In *ICWSM*, pages 401–408, 2011.
[38] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. F. Abdelzaher. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *WWW*, pages 351–360, 2017.
[39] J. Yin and J. Wang. A text clustering algorithm using an online clustering scheme for initialization. In *KDD*, pages 1995–2004, 2016.
[40] Z. Yin, L. Cao, J. Han, C. Zhai, and T. S. Huang. Geographical topic discovery and comparison. In *WWW*, pages 247–256, 2011.
[41] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *KDD*, pages 605–613, 2013.
[42] C. Zhang, K. Zhang, Q. Yuan, H. Peng, Y. Zheng, T. Hanratty, S. Wang, and J. Han. Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning. In *WWW*, pages 361–370, 2017.
[43] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han. Gmove: Group-level mobility modeling using geo-tagged social media. In *KDD*, pages 1305–1314, 2016.
[44] C. Zhang, G. Zhou, Q. Yuan, H. Zhuang, Y. Zheng, L. Kaplan, S. Wang, and J. Han. Geoburst: Real-time local event detection in geo-tagged tweet streams. In *SIGIR*, pages 513–522, 2016.
[45] L. Zhao, F. Chen, C.-T. Lu, and N. Ramakrishnan. Multi-resolution spatial event forecasting in social media. In *KDD*, 2016.
[46] L. Zhao, Q. Sun, J. Ye, F. Chen, C. Lu, and N. Ramakrishnan. Multi-task learning for spatio-temporal event forecasting. In *KDD*, pages 1503–1512, 2015.
[47] L. Zhao, J. Ye, F. Chen, C. Lu, and N. Ramakrishnan. Hierarchical incomplete multi-source feature learning for spatiotemporal event forecasting. In *KDD*, pages 2085–2094, 2016.
[48] S. Zhao, T. Zhao, I. King, and M. R. Lyu. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *WWW*, pages 153–162, 2017.