Localization of Events Using Neural Networks in Twitter Data

Usman Anjum
Dept. of Computer Science
University of Cincinnati, USA
anjumun@ucmail.uc.edu

Vladimir Zadorozhny
Dept. of Informatics and Networked Systems
University of Pittsburgh, Pittsburgh, USA
viz@pitt.edu

Prashant Krishnamurthy
Dept. of Informatics and Networked Systems
University of Pittsburgh, Pittsburgh, USA
prashk@pitt.edu

ABSTRACT

In this paper, we develop a model with neural networks to localize events using microblogging data. Localization is the task of finding the location of an event and can be done by discovering event signatures in microblogging data. We use the deep learning methodology of Bi-directional Long Short-Term Memory (Bi-LSTM) to learn event signatures. We propose a methodology for labeling the Twitter date for use in Bi-LSTM However, there might not be enough data available to train the Bi-LSTM and learn the event signatures. Hence, the data is augmented using generative adversarial networks (GAN). Finally, we combine event signatures at different temporal and spatial granularity to improve the accuracy of event localization. We use microblogging data collected from Twitter to evaluate our model and compare it with other baseline methods.

Keywords

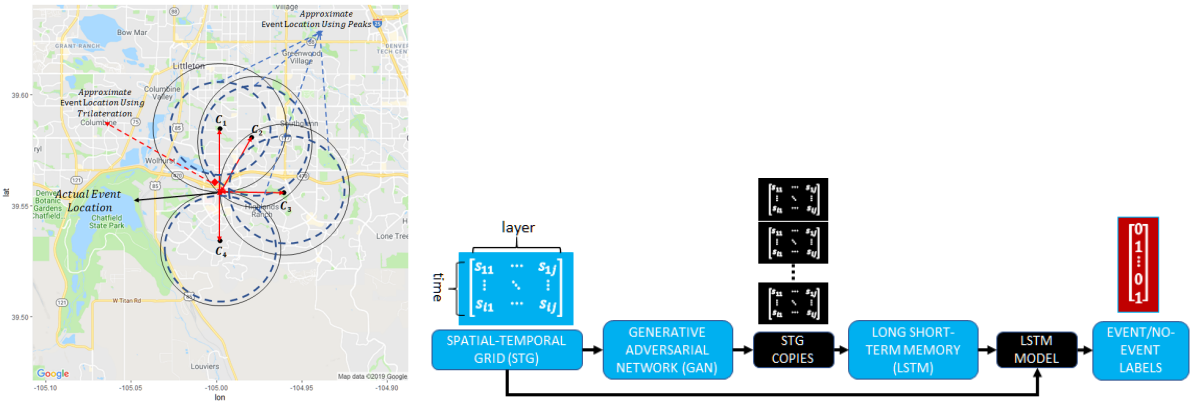
social networking, event localization, Twitter, neural networks, GAN, BiLSTM

INTRODUCTION

Localization of an event is the task of finding the location of an event. The "location" of an event could be in time or space (Anjum et al. 2022). For our model, we consider the location of an event as the geographical coordinate at which the event occurred. We define an event as a significant one-time occurrence that observes a different pattern or signature from the usual or expected behavior. We focus on rare and unexpected events with no prior knowledge about event time and location (Atefeh and Khreich 2015; Ozdikis, Oğuztüzün, et al. 2017), e.g. natural disasters (earthquakes, floods, fires, and typhoons), infectious disease outbreaks, traffic incidents, riots, shooting, and terrorist acts.

The rise in the use of various microblogging services, like Twitter, by people to share information can help us discover event patterns and we can use these patterns to find the location of an event. Finding the location of an event is not as straightforward. Microblogging data is scarce and unreliable, i.e., microblogging data is *underdeveloped* (Anjum et al. 2021), and the spatial attributes needed for localization are not reliable. Traditional event localization methods use spatial attributes found in the microblogging messages (called tweets in Twitter), like latitude and longitude (*geotags*) of the messages, location name found in the message (also called *place name*), and user location found in their profiles to localize an event (Ozdikis, Oğuztüzün, et al. 2017). The increasing usage of location anonymization techniques to hide the geotags and incorrect or multiple location names in the messages makes the information within the data unreliable which would result in an inaccurate event location (Anjum et al. 2022). The most relevant work that has considered these issues can be found in Anjum et al. 2022. In this paper, peaks were obtained from aggregated counts of tweets and the position of the peaks was used to find the location of an event. However, peaks are not easy to identify and there may be multiple peaks resulting in inaccurate results (Krumm and Horvitz 2015; Ben Lazreg et al. 2020).

In this paper, we propose the model *SPatial Event Localization (SPEL)*. SPEL is based on the scenario provided in Anjum et al. 2021 and Anjum et al. 2022. Figure 1(a) illustrates the scenario. However, unlike the previous works, we use neural networks for event localization. The scenario to perform event localization can be envisioned as users (also referred to as *social sensors* (Giridhar et al. 2015), distributed throughout a region sending out messages based on their interactions with the environment. Instead of relying on the content of the messages and the temporal and spatial attributes, we use the number of messages sent out by people within a geographic region and time window. The number or count of microblog messages are monitored by **reference points** ($C_i = (x_i, y_i)$ where x_i and y_i are geographic coordinates) within various circular regions (called *layers*) of various radii r_i in space within a time window. We can represent this information as a **Spatial-Temporal Grid (STG)** where each element represents the number of tweets within a time window at r_i from coordinates x_i and y_i. The change in the number of messages can be indicative of an event and the location can be found as lying within one of these layers. Information from multiple reference coordinates can be combined using trilateration to obtain a more exact geographic coordinate.



(a) (b)

Figure 1. (a) Event Localization Scenario (b) SPEL Model

The SPEL model is shown in Figure 1(b) In SPEL we (a) represent data as an *STG* and generate training data using a generative adversarial network (GAN) (Goodfellow et al. 2014), (b) using Bi-directional Long Short-Term Memory (Bi-LSTM) (Schuster and Paliwal 1997), (c) refine the patterns by merging obtained patterns from different location granularity and (d) eventually combine the information from multiple reference points to obtain a more accurate location.

To test our methodology we have considered mass shooting events. There are multiple reasons for doing so. Firstly, these events are identified by a geographical coordinate. Secondly, we would like to use the same event type as we believe the same event types have similar event signatures and can be used to find the location of similar events. Hence, we can use Bi-LSTM trained on an event from a different location to find the location of an event at another location.

In summary, our goals and contributions in this paper are as follows:

Formulation and Algorithm: We propose the model *SPatial Event Localization (SPEL)*. In SPEL, we propose a novel way to represent Twitter data as an STG. Next, we propose a labeling strategy to label unlabeled Twitter data. We also propose techniques to improve event localization accuracy by combining event signatures using different time and space granularity. We use generative adversarial networks (GAN) to generate data for training and Bi-directional Long Short-Term Memory (Bi-LSTM) to find the location of an event.

Accuracy: Based on our results, we are accurately able to find the location of an event as seen in Figure 2. The figure shows that the model in this paper (MAJ and AND) is able to identify the geographic coordinates that are very close to the actual event location, i.e. has low distance error.

LITERATURE REVIEW

Our literature review consists of two parts. In the first subsection, we review the literature on data augmentation. In the second subsection, we review the literature on event detection.

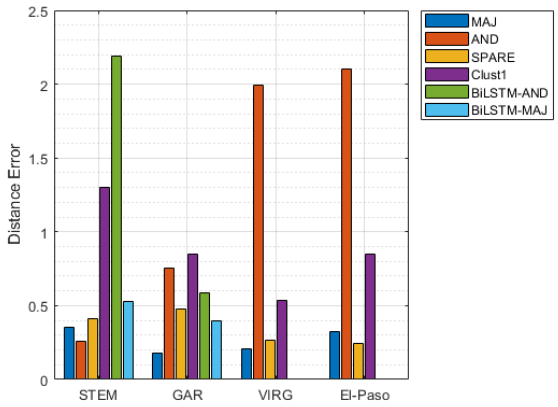


Figure 2. Summary of Results

Data Augmentation

Data augmentation has been used in previous literature for image (e.g., face data augmentation in (Wang et al. 2020), speech and natural language processing (NLP) (Dai and Adel 2020) speech and time-series data to reduce overfitting (Shorten and Khoshgoftaar 2019; Wen et al. 2020). Augmentation increases the size of the training data set by geometric and color transformations and deep learning techniques like Generative Adversarial Networks (GAN). Augmentation also alleviates the issue of class imbalance, which is a data set with skewed majority-to-minority sample ratios (Shorten and Khoshgoftaar 2019). Generative adversarial networks (GAN) were used to generate synthetic images in the medical domain (Bowles et al. 2018; Frid-Adar et al. 2018; Han et al. 2018). These works generated CT images of liver lesions (Bowles et al. 2018; Frid-Adar et al. 2018) and MR images (Han et al. 2018) which were very close in comparison to the real data. Similarly, cycle-Consistent Generative Adversarial Networks (CycleGANs) were proposed as an image classification method to detect floods using images found in social media (Pouyanfar et al. 2019).

Generating and using augmented Twitter data has not been studied extensively in the literature. An agent-based model (ABM) has been one proposed method for generating augmented data. A relevant work that used ABM for data augmentation was called Twitter Behavior Agent-Based Model (TBAM) (Anjum et al. 2021). TBAM generated Twitter data to simulate people's behavior in an event. The cross-correlation function (`ccf`) was used to validate the data generated and compare it with data collected from Twitter. However, TBAM requires a large number of parameters to be obtained from the data whereas GAN does not require any parameters to generate user data.

ABM was used to investigate how information was spread during the 2011 Wenzhou train crash through Sina Weibo (Cui et al. 2013). They use the ABM framework to compare information diffusion through word-of-mouth and mass media and to determine which is a more significant means of spreading information when it comes to social media. ABM has been used to create an information propagation model to study how retweeting occurs (Xiong et al. 2012; Pezzoni et al. 2013). In another paper, ABM was used to create a bottom-up approach to evaluate emergent behavior and identified methods to better tune their model. However, none of the models have a method for verifying the data created and do not focus on patterns in data when an event occurs (Gatti et al. 2013),.

Event Detection & Localization

There are many surveys that have summarized the work done on event detection in the microblogging domain (Atefeh and Khreich 2015; Steiger et al. 2015; Cordeiro and Gama 2016; Garg and Kumar 2016; Imran et al. 2015; Hasan et al. 2018; Ozdikis, Oğuztüzün, et al. 2017), and (Zheng et al. 2018). Each of these surveys focuses on a specific aspect of event detection. For example, one survey identified challenges and limitations arising in event detection and localization methodology due to the use of content in tweets like ambiguous texts in tweets, and lack of relevant data (Atefeh and Khreich 2015). In another survey, an advanced systemic literature review presented on methodologies and applications of Twitter as a Location-Based Social Network was presented (Steiger et al. 2015). There was also a survey focused on creating a taxonomy of event detection in social media and the different methods are classified under type of event, type of detection method (supervised or unsupervised), and if the event detected is a new event or an old event (Cordeiro and Gama 2016). Garg and Kumar focused on the different types of data sets (images, texts, audio, etc.) in social media used for event detection. The survey by Imran et al. (Imran et al.

2015) covers approaches, the challenges, and the benefits of different approaches for use of social media messages for detecting emergency events (like natural disasters, etc.). Finally, Hasan et al. (Hasan et al. 2018) a survey on methods for real-time detection of events is done.

The survey by Ozdikis, Oğuztüzün, et al. (Ozdikis, Oğuztüzün, et al. 2017) studied a list of techniques proposed for event localization in Twitter, classifying them based on spatial features used for location estimation and granularity of location estimation. They looked at unsupervised methods and spatial clustering techniques, probabilistic techniques, and metrics used for location estimation. Similarly, Zheng et al. (Zheng et al. 2018) explores the geo-location problem and challenges associated with finding the home location, tweet location, and the mentioned location. They show that the methods used rely heavily on the tweet content and the noisy and short nature of the tweets makes geo-location a challenge.

The location of an event can be found as a single geographical coordinate, geographic area, or a name identified by users in their tweets (Ozdikis, Oğuztüzün, et al. 2017). In Abdelhaq et al., the region of interest is divided into cells and then keywords are extracted based on their temporal and geo-spatial properties and then clustered. A cluster is said to be a localized event if its keywords have a high frequency, is a member of a cluster for a long time, and was recently bursty in the same cluster. The *Eyewitness* algorithm (Krumm and Horvitz 2015), and its real-time version (Comito et al. 2017), looks through a corpus of geotagged tweets over localized regions for unusual spikes in tweet counts. They divide the area of interest into triangles and use time periods of different lengths. An event is defined as a peak above a baseline tweet count, which is obtained through regression. However, during pre-processing, they remove retweets and repeat tweets which, we believe, may play a significant role in event detection. Furthermore, they do not discuss spurious peaks which we show later can cause inaccuracies in results. A geo-social event detection method focusing on the geographical regularities of local crowd behaviors to detect events has also been proposed (Lee and Sumiya 2010). They implemented their method using a fixed time window and their geographic grids are created based on a clustering-based space partition method.

A framework called *SPatial Aggregation REconstruction (SPARE)* considered peaks to find the location of an event (Anjum et al. 2022). The geographic region was divided into concentric circles. The peaks were obtained when the number of tweets measured at specific radii from the center of the circle was disaggregated. The peaks were obtained from multiple reference coordinates and then the information was combined using trilateration to obtain a more exact geographic coordinate. Low pass filters were also used to remove any unwanted random peaks. A more advanced version of a filter to remove the random peaks, called Semantic Decay Filter (SDF), was proposed (Ben Lazreg et al. 2020). The SDF removes peaks that have low similarity between texts in tweets. In another work (Cheng and Wicks 2014), clusters are created in space and time. Then the clusters are classified based on the topics within each cluster.

Sakaki et al. (Sakaki et al. 2010) used tweets to find the epicenter of an earthquake and the trajectory of typhoons. First, semantic analysis of the texts in the tweets is done to extract the relevant tweets. The authors assume that tweets follow an exponential distribution with time which is used to estimate the probability of the occurrence of an event. Next, they use the tweets' geographic coordinates to estimate an event's location and trajectory using Kalman filters and particle filters. Kalman filters assume a Gaussian distribution of the coordinates and particle filters look at how the users are distributed in a region. Another work (Ozdikis, Oguztuzun, et al. 2013) estimates an event's location by assigning probabilities using Dempster–Shafer (DS) theory based on geotags, texts in tweets, and user profiles. The location of the events was found by clustering. However, they only considered two levels of granularity and require coordinates and names for assigning probabilities. This work was extended to incorporate real-time tweets (Ozdikis, Oğuztüzün, et al. 2016). Dempster–Shafer (DS) was also used to find the coarse-grained information (like city name) and fine-grained information (coordinates of the event) (Shahraki et al. 2019). They focused mostly on traffic accidents.

METHODOLOGY

In this section, we describe our SPEL Model (see Figure 1(b)). The first stage involves generating an *STG* matrix. Using a GAN, we generate multiple copies of the *STG* matrix. The GAN-generated data is used to train the Bi-LSTM. The Twitter data is unlabeled, so we propose a labeling strategy for training the Bi-LSTM. Next, we test our model on data collected from the real world to obtain a rough estimate of the location of the event as a radius around a coordinate. Finally, we perform localization by combining information to find a more precise geographic location.

Spatial Temporal Grid (STG) for Twitter Data

In this paper, we propose using number of tweets (or counts) of tweets to find event signatures rather than using contents and messages (Anjum et al. 2022). However, instead of disaggregating the counts of tweets and using the

peaks to find the location of the events, we will instead use the aggregated data to train the Bi-LSTM model to find the location of events.

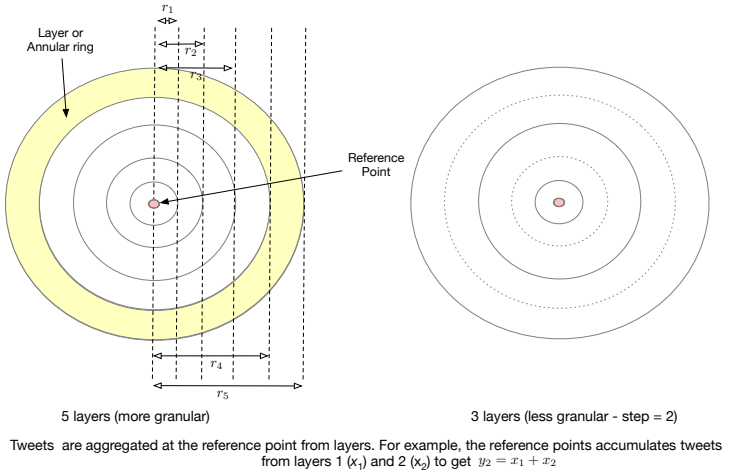


Figure 3. Generating a Spatial-Temporal Grid (STG)

As shown in Figure 3, the number of tweets can be envisioned as lying in a circle or layer at a fixed distance (radius) from a reference point measured within a time window. The number of tweets is the total number of *all the tweets within the circle*, e.g. at r_5 the number of tweets are the sum of tweets observed at r_1, r_2, r_3, and r_4. We can represent this as a matrix which we refer to as the **spatial-temporal grid (STG)** (Equation 1).

$$STG = \begin{bmatrix} s_{11} & \cdots & s_{1j} \\ \vdots & \ddots & \vdots \\ s_{i1} & \cdots & s_{ij} \end{bmatrix} \tag{1}$$

where i is the total number of layers and j is the total time over which the number of tweets was measured.

Training Data Generation

In the previous section, we defined the STG. Each reference point C_i would have a unique STG. Within this STG we can find an event or non-event patterns. These patterns within the STG are hidden or *implicit* patterns as they may not be obvious, unlike peaks which could be considered obvious or *explicit* patterns. We believe that using a deep learning method like Bi-LSTM would be the best way to find such implicit patterns It is commonly observed that peaks may occur well after the event has occurred and hence, the implicit patterns would be a better indication of an event rather than using peaks.

However, in order to train Bi-LSTM, we would require significant data. We show later that using only limited data obtained directly from the events may not identify signatures. Hence, we propose to use GAN to generate multiple copies of the STG (also referred to as augmentation). Our inspiration for using GAN comes from image processing where images are presented as 2-dimensional matrices and to improve the classification accuracy, GAN is used to generate copies of the images, e.g. in image recognition (Han et al. 2018). Consequently, we use a simple GAN architecture proposed by Goodfellow et al. to generate multiple copies of the STG which serves as the training data. The generated STG copies are denoted $\hat{STG}_i$ where i is from 1 to n.

Using the GAN-generated data, we can now train the Bi-LSTM. However, the data is unlabeled, and to use STG copies for training the Bi-LSTM model, the training data needs to be assigned a label. We propose a simple strategy to assign labels, called `Pre-Post Labeling (PPL)`.

The PPL strategy can be illustrated using Figure 4. It should be noted that in the training data, the time and location of the event are known. Hence, using this information each $\hat{STG}_i$ is split at the row at which the event occurred (denoted as k in Figure 4) into pre-event (shown as the blue matrix in Figure 4) and post-event matrices (shown as the red matrix in Figure 4). For each pre- and post-event matrix, δ layers before the event are considered non-event signatures, and δ layers after the event are considered event signatures. The non-event signature is given a label of 0 and the event signature which is observed after the event is given a label of 1. Each slice before event is denoted $(\hat{STG}_i)^{pre}_{\delta}$ and layer after the event is denoted $(\hat{STG}_i)^{post}_{\delta}$. Using different window slices δ, multiple slices can be created before and after the event. This data serves as the training data for Bi-LSTM. Different δ values are used to train multiple Bi-LSTM models.

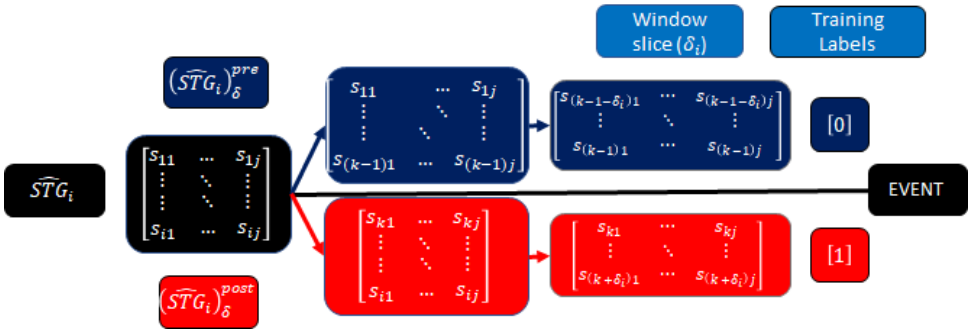
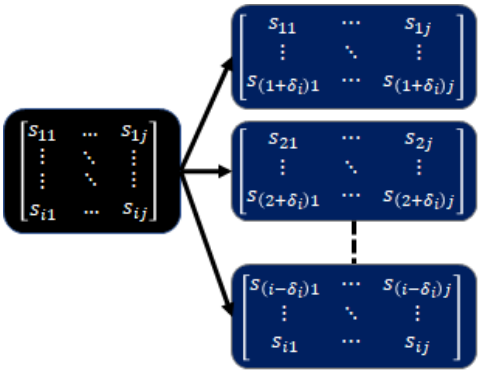


Figure 4. Methodology for Assigning Training Labels - `Pre-Post Labeling (PPL)`

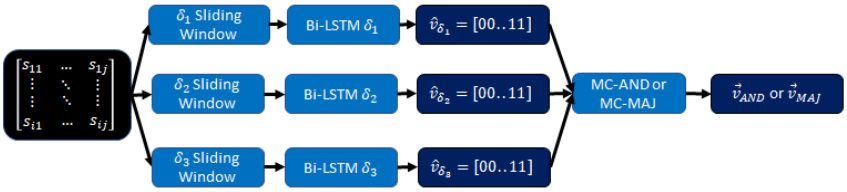
Testing Data Generation and Procedure

Using the trained Bi-LSTM model, the next task is to find event and non-event signatures in data that has been obtained from real-world events using Twitter. We envision a scenario, where there will be a stream of tweets that will represent the number of tweets within a time window at layers of size δ. Figure 5(a) shows how the STG is divided into slices of size δ for testing the Bi-LSTM.

Figure 5(b) illustrates the testing process for three different δ values (δ_1, δ_2 and δ_3). The STG is divided into sliding windows of different δ values. The data is tested on the Bi-LSTM model trained using the data from PPL method with the corresponding δ. As the event is unknown, the event signatures would be hidden within the slices of the STG. For every slice, if an event signature has been detected the label of 1 is assigned and a label of 0 is assigned when no event signature is detected. In this way, a label vector $\hat{v}_\delta$ is obtained. It should be noted that at different δ values, the event signatures may be different and different $\hat{v}_\delta$ label vectors are obtained.



(a) Dividing STG into time slice (window) of size δ



(b) Obtaining event vector

Figure 5. Testing Process

The multiple $\hat{v}_\delta$ can be combined for a more accurate event/no-event signature label vector. For combining the label vectors, we propose the method called `MAJ Combine (MC)`. In `MC`, a new matrix, $\boldsymbol{V_\delta}$, is created by taking δ repetitions of each element in $\hat{v}_\delta$ shifted by 1. It should be noted that the number of rows of $\boldsymbol{V_\delta}$ is equal to the length of $\hat{v}_\delta$. Depending on how we combine $\hat{v}_\delta$, `MC` splits in the following two methods:

- `MC-AND`: *AND* of each element in the column of $\boldsymbol{V_\delta}$ to generate $\vec{v}_{\delta-AND}$. The method is repeated for different δ values to generate multiple $\vec{v}_{\delta-AND}$. Then we take the logical `AND` of all the $\vec{v}_{\delta-AND}$ to obtain $\vec{v}_{AND}$.

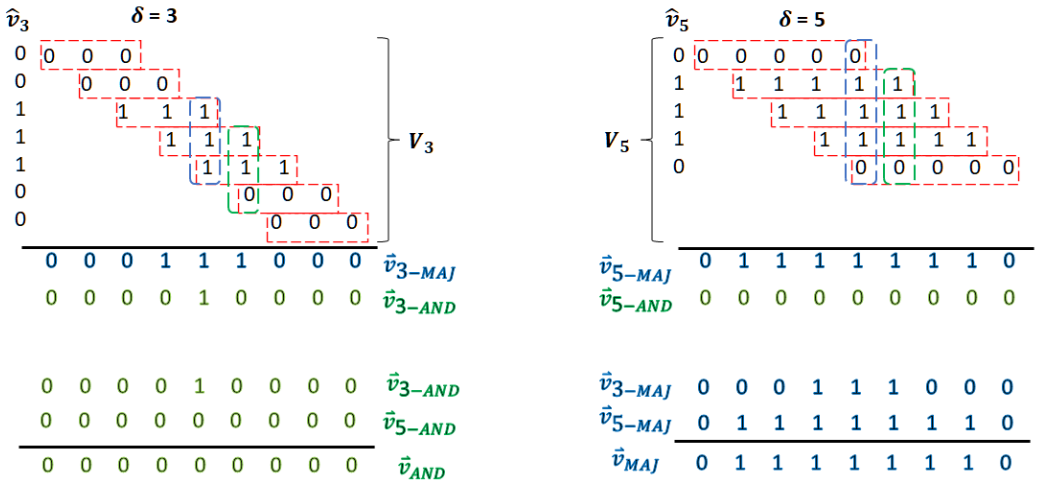


Figure 6. `Matrix Combine (MC)` Example with 2 different δ

- `MC-MAJ`: In this method we first count the number of $0s$ and $1s$ in the columns of $\boldsymbol{V_\delta}$ to create a label vector $\vec{v}_{\delta-MAJ}$. If there are more $0s$ than $1s$ in the column then a 0 is assigned and if there are more $1s$ than $0s$ then a 1 is assigned to that position in $\vec{v}_{\delta-MAJ}$. The method is repeated for different δ values to obtain multiple $\vec{v}_{\delta-MAJ}$. We implement majority voting again on all the $\vec{v}_{\delta-MAJ}$ to obtain $\vec{v}_{MAJ}$.

Figure 6 shows an example for the `MC` methods shows an example for the MC method using two δ values, $\delta = 3$, and $\delta = 5$, and how they are combined together. In this way, there are two different binary label vectors $\vec{v}_{AND}$ and $\vec{v}_{MAJ}$ that show the event/no-event signatures.

Event localization and trilateration

In the previous section, we used the real-world data on the trained Bi-LSTM model to obtain vectors $\vec{v}_{AND}$ and $\vec{v}_{MAJ}$. These vectors are obtained for a single reference coordinate and there would be multiple vectors obtained for each of the reference coordinates. In this section, we describe how the vectors from the previous section can be used to localize an event. It should be noted that the 1 in $\vec{v}_{AND}$ and $\vec{v}_{MAJ}$ represent the layers at which the event occurred. It has been observed from previous works (Anjum et al. 2022) that the event signature would usually manifest along contiguous layers. That is, the influence of the event on people closest to the event would be strongest and would remain strong for a specific distance. Hence, the longest contiguous $1s$ would be the event signature. We propose that the layer or radius at which the event occurred would be the position of the first 1 of the longest contiguous $1s$ in the vectors $\vec{v}_{AND}$ and $\vec{v}_{MAJ}$.

Consequently, for each of the reference coordinates (C_i), we would identify a specific layer as a radius at which the event occurred. To get the exact latitude and longitude, we combine the different layers for each of the reference coordinate to get an exact latitude and longitude (instead of only a radius), as was illustrated in Figure 1(a). We use the concept of trilateration (Dargie and Poellabauer 2010) for combining the radius from the different C_i. Previous works used peaks to identify the radius, whereas we used Bi-LSTM to find the radius. Trilateration has been used widely in sensor localization to find the location of an unknown sensor based on its distance from sensors at fixed known locations. Next, we describe the implementation of trilateration for finding the geographic coordinate.

Let there be n reference coordinates, C_i where $i = 1....n$ (called anchors nodes in trilateration), whose coordinates are represented in the 2D Cartesian plane as (x_i, y_i) where $i = 1....n$. The unknown coordinate (which is the possible event location) is represented by coordinates $\mathbf{x} = (x, y)$. The distance between the approximate event location and the reference coordinates is the layer at which the significant peak lies. It is denoted r_i for reference coordinate C_i respectively. The relationship between sensor nodes, approximate events, and distances is represented as $A\mathbf{x} = b$, where:

$$A = \begin{bmatrix} 2(x_n - x_1) & 2(y_n - y_1) \\ 2(x_n - x_2) & 2(y_n - y_2) \\ \vdots & \vdots \\ 2(x_n - x_{n-1}) & 2(y_n - y_{n-1}) \end{bmatrix}$$

$$b = \begin{bmatrix} {r_1}^2 - {r_n}^2 - {x_1}^2 - {y_1}^2 + {x_n}^2 + {y_n}^2 \\ {r_2}^2 - {r_n}^2 - {x_2}^2 - {y_2}^2 + {x_n}^2 + {y_n}^2 \\ \vdots \\ {r_{n-1}}^2 - {r_n}^2 - {x_{n-1}}^2 - {y_{n-1}}^2 + {x_n}^2 + {y_n}^2 \end{bmatrix}$$

Using least square estimation, $\mathbf{x}$ can then be found using $\mathbf{x} = (A^T A)^{-1} A^T b$.

For 2 reference coordinates the trilateration formula is slightly different. Using geometry, the coordinates can be found using the formula:

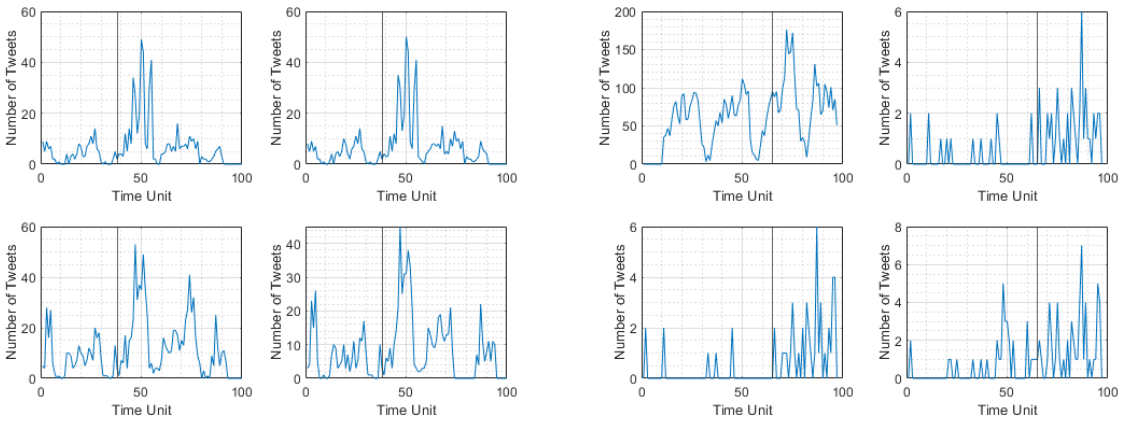
$$\begin{aligned} x &= \frac{r_1^2 - r_2^2 + D}{2D} \\ y &= \pm\sqrt{r_1^2 - x^2} \end{aligned} \tag{2}$$

where D is the distance between C_1 and C_2 and $D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

EXPERIMENTS AND RESULT

In this section, we use the real-world dataset to look at the performance of our model.

Dataset



(a) STEM Dataset Collected from 4 Reference Points

(b) VIRG Dataset Collected from 4 Reference Points

Figure 7. Changing Number of Tweets with Time at the Outermost Layer

The data for analysis was obtained from TBAM (Anjum et al. 2021) and SPARE (Anjum et al. 2022) models [1]. The description of the datasets is summarized in Table 1. The table shows the event name, the reference coordinates at which the number of tweets was collected, the actual coordinate of the event, and the time at which the event occurred. The *Label* refers to the name used for the data sets in the experiments. Each of the events is an example of an unknown event. For each reference coordinate, the number of tweets was collected from a radius of 1.0mile to 2.8miles with 0.1 mile increments. The number of tweets is measured from 2 days before the event to 2 days after the event occurrence data with a 1-hour time step. The STG would have $i = 2.8$ and $j = EventData + 2days$. Figures 7(a) and 7(b) shows the changing number of tweets at the outermost layer (which is at 2.8miles) collected from the 4 different reference points for the STEM and VIRG datasets respectively.

[1] https://github.com/usmananjum/SPARE_Data.git

Table 1. Summary of Real Data

Data Set	Reference Coordinate (Latitude, Longitude)	Label	Event Date & Time	Event Coordinate (Latitude, Longitude)
STEM School Shootings	39.58482, −104.99790	STEM	05-07-2019 1:53pm	39.556, −104.9979
	39.58096, −104.97928			
	39.55599, −104.96067			
	39.53438, −104.99790			
Virginia Beach Shootings	36.75089, −76.02167	VIRG	05-31-2019 4:44pm	36.7509, −76.0575
	36.75089, −76.02167			
	36.72206, −76.05750			
	36.75089, −76.09333			
Garlic Festival Shootings	37.02661, −121.58528	GAR	07-28-2019 5:40pm	36.997778, −121.585278
	36.99777, −121.54933			
	36.96894, −121.58528			
	36.99777, −121.62123			
El-Paso Shootings	31.80596, −106.38430	ELP	09-03-2019 10:45am	31.7771, −106.3843
	31.7771, −106.3505,			
	31.74824, −106.38430			
	31.7771, −106.4096			

Results

We created the GAN network that was implemented in TensorFlow in R. The Bi-LSTM was implemented using Matlab with 200 layers and was trained for 5 epochs. Three different δ values were used: $\delta = 2, 4$ and 6. We first generated data with GAN using $STEM$, $VIRG$, and ELP datasets. Using this GAN-generated data we trained the Bi-LSTM. Next, we tested the Bi-LSTM model on the $STEM$, $VIRG$, GAR, and ELP datasets. The GAR datasets acted as the unknown event that was not part of the training data. The results of our experiments are in Table 2 and bar plot in Figure 2. In the table, $SPEL-MAJ$ and $SPEL-AND$ are the models proposed in this paper that implemented $MC-MAJ$ or $MC-AND$ to combine the different δ values respectively. $LSTM-MAJ$ and $LSTM-AND$ are the baseline models that use Bi-LSTM trained without GAN data and using only the event data. Similar to $SPEL-MAJ$ and $SPEL-AND$, $BiLSTM-MAJ$ and $BiLSTM-AND$ uses $MC-MAJ$ or $MC-AND$ to combine the different δ values respectively. $SPARE$ (Anjum et al. 2022) is the state-of-the-art method we compare our work to. $kmeans$ is a $kmeans$ clustering algorithm that used the content of the tweets to cluster the tweets. The center of the cluster is the approximate event coordinate.

In the table, we report the distance error. The distance error is the difference between the actual event coordinate and the event coordinate that is estimated by different methods and was reported in miles. The distance error is found as the distance between the points on the sphere or ellipsoid.

The results show that $SPEL$ outperforms all the methods in $STEM$, $VIRG$, and GAR datasets. Our method has a slightly higher distance error than the $SPARE$ method for the ELP dataset, but it outperforms the baselines by a significant margin. The $BiLSTM-AND$ and $BiLSTM-MAJ$ in some instances are not even able to identify event signatures from any of the reference coordinates (as denoted by *NA*). On the other hand, $SPEL$ is able to identify signatures from at least 1 reference coordinate. This shows that using GAN-generated data to train Bi-LSTM would yield much better performance. Furthermore, $SPEL$ is successfully able to detect events in GAR with very good accuracy even though GAR was not in the original training set. This shows that our model finds the location of similar events that occur at different locations.

Table 2. Summary of Results - Error in Distance (in miles)

Method	STEM	VIRG	GAR	ELP
SPEL-MAJ	0.3566	**0.2598**	0.2308	*0.2801*
SPEL-AND	**0.3084**	0.7549	**0.1470**	1.9941
SPARE	0.4083	0.4743	0.2678	**0.2422**
kmeans	1.2980	0.8488	0.5320	0.8486
BiLSTM-MAJ	2.2988	0.4101	1.9836	NA
BiLSTM-AND	2.1628	NA	NA	NA

Conclusion

This paper presents a novel model called *SPatial Event Localization (SPEL)*. *SPEL* uses Bi-LSTM to localize an event. The Bi-LSTM was trained using GAN-generated data. Our model is able to perform better than the baseline models. The results show that SPEL can localize an event with limited training data availability and can also localize events whose data was not found in the training set. There are limitations to our model. We used a simple GAN and Bi-LSTM model for localization which meant that sometimes the result was not very good and in some cases, the reference coordinates did not report any event. We believe by exploring other classification models, like Bayesian models, we could achieve better results. We could also explore other augmentation models like autoencoders and diffusion models (Luo 2022) to generate more realistic data. Research could also be done to measure the quality of the generated data and how real is the generated data. In addition, we assumed that each of the *STG* had only a single event information in it. It would be interesting to create a model that can localize multiple events. We used a single type of event, the shooting event, to localize the event. In the future, we also want to study event signatures of other types of events (instead of only shooting events) and study the correlation between different events and see how different event types could be used for localizing unknown events.

REFERENCES

Abdelhaq, H., Sengstock, C., and Gertz, M. (2013). "Eventweet: Online localized event detection from twitter". In: *Proceedings of the VLDB Endowment* 6.12, pp. 1326–1329.

Anjum, U., Zadorozhny, V., and Krishnamurthy, P. (2021). "TBAM: Towards An Agent-Based Model to Enrich Twitter Data". In: *18th ISCRAM Conference Proceedings. Blacksburg, VA (USA): Virginia Tech.* ISCRAM.

Anjum, U., Zadorozhny, V., and Krishnamurthy, P. (2022). "Localization of Unidentified Events with Raw Microblogging Data". In: *Online Social Networks and Media* 29, p. 100209.

Atefeh, F. and Khreich, W. (2015). "A survey of techniques for event detection in twitter". In: *Computational Intelligence* 31.1, pp. 132–164.

Ben Lazreg, M., Anjum, U., Zadorozhny, V., and Goodwin, M. (2020). "Semantic decay filter for event detection". In: *17th ISCRAM Conference Proceedings. Blacksburg, VA (USA): Virginia Tech.* ISCRAM, pp. 14–26.

Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D. A., Hernández, M. V., Wardlaw, J., and Rueckert, D. (2018). "Gan augmentation: Augmenting training data using generative adversarial networks". In: *arXiv preprint arXiv:1810.10863*.

Cheng, T. and Wicks, T. (2014). "Event detection using Twitter: A spatio-temporal approach". In: *PloS one* 9.6, e97807.

Comito, C., Falcone, D., and Talia, D. (2017). "A peak detection method to uncover events from social media". In: *Data Science and Advanced Analytics (DSAA), 2017 IEEE International Conference on*. IEEE, pp. 459–467.

Cordeiro, M. and Gama, J. (2016). "Online social networks event detection: a survey". In: *Solving Large Scale Learning Tasks. Challenges and Algorithms*. Springer, pp. 1–41.

Cui, K., Zheng, X., Zeng, D. D., Zhang, Z., Luo, C., and He, S. (2013). "An empirical study of information diffusion in micro-blogging systems during emergency events". In: *International Conference on Web-Age Information Management*. Springer, pp. 140–151.

Dai, X. and Adel, H. (2020). "An Analysis of Simple Data Augmentation for Named Entity Recognition". In: *arXiv preprint arXiv:2010.11683*.

Dargie, W. and Poellabauer, C. (2010). *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. (2018). "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification". In: *Neurocomputing* 321, pp. 321–331.

Garg, M. and Kumar, M. (2016). "Review on event detection techniques in social multimedia". In: *Online Information Review* 40.3, pp. 347–361.

Gatti, M., Cavalin, P., Neto, S. B., Pinhanez, C., Santos, C. dos, Gribel, D., and Appel, A. P. (2013). "Large-scale multi-agent-based modeling and simulation of microblogging-based online social network". In: *International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Springer, pp. 17–33.

Giridhar, P., Abdelzaher, T., George, J., and Kaplan, L. (2015). "Event localization and visualization in social networks". In: *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, pp. 35–36.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680.

Han, C., Hayashi, H., Rundo, L., Araki, R., Shimoda, W., Muramatsu, S., Furukawa, Y., Mauri, G., and Nakayama, H. (2018). "GAN-based synthetic brain MR image generation". In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, pp. 734–738.

Hasan, M., Orgun, M. A., and Schwitter, R. (2018). "A survey on real-time event detection from the twitter data stream". In: *Journal of Information Science* 44.4, pp. 443–463.

Imran, M., Castillo, C., Diaz, F., and Vieweg, S. (2015). "Processing social media messages in mass emergency: A survey". In: *ACM Computing Surveys (CSUR)* 47.4, p. 67.

Krumm, J. and Horvitz, E. (2015). "Eyewitness: Identifying local events via space-time signals in twitter feeds". In: *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, p. 20.

Lee, R. and Sumiya, K. (2010). "Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection". In: *Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks*. ACM, pp. 1–10.

Luo, C. (2022). "Understanding diffusion models: A unified perspective". In: *arXiv preprint arXiv:2208.11970*.

Ozdikis, O., Oguztuzun, H., and Karagoz, P. (2013). "Evidential location estimation for events detected in twitter". In: *Proceedings of the 7th Workshop on Geographic Information Retrieval*. ACM, pp. 9–16.

Ozdikis, O., Oğuztüzün, H., and Karagoz, P. (2016). "Evidential estimation of event locations in microblogs using the Dempster–Shafer theory". In: *Information Processing & Management* 52.6, pp. 1227–1246.

Ozdikis, O., Oğuztüzün, H., and Karagoz, P. (2017). "A survey on location estimation techniques for events detected in Twitter". In: *Knowledge and Information Systems* 52.2, pp. 291–339.

Pezzoni, F., An, J., Passarella, A., Crowcroft, J., and Conti, M. (2013). "Why do I retweet it? An information propagation model for microblogs". In: *International Conference on Social Informatics*. Springer, pp. 360–369.

Pouyanfar, S., Tao, Y., Sadiq, S., Tian, H., Tu, Y., Wang, T., Chen, S.-C., and Shyu, M.-L. (2019). "Unconstrained Flood Event Detection Using Adversarial Data Augmentation". In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 155–159.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). "Earthquake shakes Twitter users: real-time event detection by social sensors". In: *Proceedings of the 19th international conference on World wide web*. ACM, pp. 851–860.

Schuster, M. and Paliwal, K. K. (1997). "Bidirectional recurrent neural networks". In: *IEEE transactions on Signal Processing* 45.11, pp. 2673–2681.

Shahraki, Z. K., Fatemi, A., and Malazi, H. T. (2019). "Evidential fine-grained event localization using Twitter". In: *Information Processing & Management* 56.6, p. 102045.

Shorten, C. and Khoshgoftaar, T. M. (2019). "A survey on image data augmentation for deep learning". In: *Journal of Big Data* 6.1, p. 60.

Steiger, E., De Albuquerque, J. P., and Zipf, A. (2015). "An Advanced Systematic Literature Review on Spatiotemporal Analyses of T witter Data". In: *Transactions in GIS* 19.6, pp. 809–834.

Wang, X., Wang, K., and Lian, S. (2020). "A survey on face data augmentation for the training of deep neural networks". In: *Neural Computing and Applications*, pp. 1–29.

Wen, Q., Sun, L., Song, X., Gao, J., Wang, X., and Xu, H. (2020). "Time Series Data Augmentation for Deep Learning: A Survey". In: *arXiv preprint arXiv:2002.12478*.

Xiong, F., Liu, Y., Zhang, Z.-j., Zhu, J., and Zhang, Y. (2012). "An information diffusion model based on retweeting mechanism for online social media". In: *Physics Letters A* 376.30-31, pp. 2103–2108.

Zheng, X., Han, J., and Sun, A. (2018). "A survey of location prediction on twitter". In: *IEEE Transactions on Knowledge and Data Engineering* 30.9, pp. 1652–1671.