2021 6th International Conference for Convergence in Technology (I2CT)
Pune, India. Apr 02-04, 2021

Similar Event Detection and Event Topic Mining in Social Network Platform

Pramod Bide[1], Sudhir Dhage[2]
[1,2]Sardar Patel Institute Of Technology, Andheri
Mumbai, Maharashtra, India
[1]pramod_bide@spit.ac.in
[2]sudhir_dhage@spit.ac.in

2021 6th International Conference for Convergence in Technology (I2CT) | 978-1-7281-8876-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/I2CT51068.2021.9418110

***Abstract*—Social media is widely used to share information globally and it also aids to gain attention from the world. When socially sensitive incidents like rape, human rights march, corruption, political controversy, chemical attacks occur, they gain immense attention from people all over the world, causing microblogging platforms like Twitter to get flooded with tweets related to such events. When an event evolves many other events of a similar nature have happened in and around the same time frame. These are similar events because they are linked to the nature of the main event. Discussions in state-of-the-art papers are restricted to event detection, user interest discovery and detecting influential spreaders but overlook similar event detection and topic evolution. This leads to difficulty in tracing the nature of evolving events. Hence, similar event detection is critical in determining the nature of events. Similar events have fulcrums points, i.e.,topics around which the discussion is focused, as the event evolves which must be considered in topic evolution. We have proposed Event Detection model which detects similar events that are similar with regards to their temporal nature resulting from main events. The model also considers event topics, user supremacy index to calculate sub event detection factor (α). A self-tuning clustering algorithm is proposed to combine tweets, forming clusters which are composed of key posts with similar context and hence, similar topics. The sub event detection algorithm reveals events that overlap in both time and context to evaluate the effects of these similar events on deliberate human actions. The topic evolution algorithm puts into perspective the change in topics for an event's lifetime. The experimental results on a real Twitter data set demonstrate the effectiveness and precision of our proposed model for similar sub event detection during the evolution of similar events .**

***Index Terms*—Event Detection, Sub event detection, Twitte, STCA, Event Topic Mining**

I. Introduction

Social networks are a crucial part of the daily life of all users in the world. Thousands of events occur on the social network and users participate in these events by sharing, retweet, tweet, commenting, blogging over social media platforms. Users share their views and opinions leading to the continuous evolution of these events. [1]

These concurrently transpiring events over the social network gather research attention in pursuance for detecting evolving events, discovering user interest, detecting influential spreaders, detecting rumors, etc. [2]–[11] Twitter is a commonly used microblogging platform where users share short messages of 140 characters called 'tweets' that are event specific [12]. Social networking sites in recent years play a vital role in spreading evolving events, including hoaxes and other misleading messages, rapidly. Social media has become the primary cause of concern in managing man-made disasters due to its high tendency of spreading rumors.

Man Made Disaster (MMD) [13] [14] are considered to be more dangerous as compared to natural disasters when it comes to spreading information about some crisis over social media platforms [15]. MMD includes activities like rape, protests, human rights march, terrorist attacks, reservation (Aarakshan). On social media platforms, posts involve criticism, opinions, judgment, and then spreading related but different posts diverting the flow of the topic. Such posts create disputes concerning our main topic. It may happen that when one topic is evolving, it may lead to a series of events, some of which may be similar. Similar events are evolving events that are similar to that of the main event and happen within the same time frame. When a particular event happens, social networks are flooded with abundant related tweets. Taking the example of pertaining Rape case #JusticeForAsifa, when this event started evolving a few more events came into light, like Candle March, Protests, #JusticeForGeeta, and so on. During event evolution, multiple subevent follow. Events, #JusticeForAsifa and #JusticeForGeeta are similar and occurred in a close period making it imperative to detect similar events concerning root events. Existing research deals with key users related to a particular event [16] [17], whereas here we are not only considering key users but also the similar topics of each user concerning similar events. We aim at detecting such two events analyzing the effects of such negatively correlated similar events to understand their contribution to the causes leading to MMD.

When an event is trending i.e., gains high popularity, then we have observed the manifestation of sub-events branching from the root event. During event evolution, users share their thought apropos to the main event due to either their curious nature or to simply share the truth. Users may also end up making contradictory statements or offering condolences. At times, it is simply to gain attention [18]. Correspondingly, an event and its sub-events revolve around varied fulcrum points through the course of its lifetime necessitating the tracking of topic evolution during similar event evolution.

978-1-7281-8876-8/21/$31.00 ©2021 IEEE

Government officials may use this information to analyze the nature of evolving similar events to understand in what context the events are evolving and easily identify the severity of these events [19].

Given the entire event data that is produced during event evolution, similar event detection and topic evolution algorithms help to track and detect the variations in topics contained in the similar events. Recent research focuses on event detection methods only overlooking similar event detection and topic evolution. On that account, it is important to detect similar events and track evolving topics.

Detecting the evolution of events and their connection with the associated sub-events is essential to codify the fundamental knowledge of how the main event contributes to the evolution of the topic and also to recognize the compelling events. Such events are described as a cluster of countless keywords when identified using the most current techniques. The process of recognizing events, categorizing them, and summarizing such events is conducted manually. These techniques lacked the effectiveness of identifying important incidents. Also, current topic models when implemented directly to the microblogging platform result in low-quality recognition [**?**] due to information insufficiency issues. To solve this problem, we propose a Similar Event Evolution Detection model named SEDM. In our proposed model, we first identify the topics and detect such events using our Topic Decision based Event Detection Algorithm. This algorithm uses LDA [20] with modified parameters to cluster the microblogs which help to find the critical events. All events evolve and it is necessary to understand the flow of events over time. This is addressed in our model through the means of the Topic Evolution Algorithm (TEA). The main contributions of this paper are listed as follows:

1) On microblogging platforms like Twitter it is essential to understand the effects the microblogs will have on society considering the various events of similar nature occurring in the same time frame. We propose a sub-event detection algorithm. Sub event detection algorithm model considers various parameters as listed in Table III. Similar sub-events are those two events that occur in the same timespan. Such events are related to the same topic but are not necessarily similar. Taking the example of #JusticeForAsifa and #JusticeForGeeta, even though both events lie under the same area of the topic they are similar.
2) Once the topic's evolution is analyzed, one should also be able to learn the evolution pattern and predict the possible trailing events. We propose a novel Topic Evolution Algorithm (TEA). It works in two phases which helps in identifying the topics related to one fulcrum point and predict potential future events as the topic evolves. In Phase 1, TEA focuses on how the topic evolves as the context of event topics may change over time. There is a possibility that a particular topic that is the key in the initial stage may not continue to be the key in the final stage. TEA algorithm is designed such that it discovers the changes in the progression of corporeality in the event cluster to recognize similarities among the keywords that are in focus as the time window shifts through the event lifespan. This algorithm considerably increases the effectiveness and precision of event evolution. For example, the event #JusticeForAsifa further evolved into protests, candle marches, and also inflamed Religious tension in India. Now as we have analyzed how the topics evolve in our Phase 1, we move towards understanding the evolution pattern and predict what could be the further events based on the previous knowledge in Phase 2, we use the concept of collaborative filtering to predict the possible trailing events using previous knowledge. This can help us prevent the critical effects of MMD.
3) Experimental results carried out on the Twitter microblogging platform. The findings after results indicate that the sub-event detection algorithm offers comprehensive analytical data on all important evolving events. This shows the effectiveness and precision of our proposed model for both similar-event detection and topic evolution.

The rest of this paper is organized as follows: First, in Section II, we discuss related research for event detection and event evolution on microblogging platforms. Then, in Section III, we present our SEDM model. Finally, we present our experimental results in Section IV. We conclude our research in the last Section V.

II. Related Work

Topic Modelling is used to detect various flooding events on a social network. Probabilistic Latent Semantic Analysis (PLSA) [21] has gained a lot of attention for detection and keeping track of evolving hot events. [7]–[10] mention how PLSA is a statistical technique for analyzing the cooccurrence of data along with finding hidden variables from observed variables. Latent Dirichlet allocation (LDA) [20]] is a generative statistical model allowing sets of observations to be explained by unobserved groups by identifying a set of topics belonging to one document. For finding similarities, these three models are widely used. Many states of the art papers only discuss topic detection and not event evolution with time, change in events leading to the creation of new events. Also, it ignores how they are cross related to each other. Evolving topics and focus on the prediction of upcoming events needs to be considered.

A fuzzy represented and timely evolved theoretic information matrix for Twitter dynamics was proposed by Nikolaos D. Doulamis [22] [23] [9], though the accuracy achieved was not up to the mark. Twitter-based detected events data sets, [4] online event detection approach is utilized in combination with word embedding but they fail to detect cross events. A hot event evolution model that considered short text data in social networks with users, interest distribution is proposed by Lei-Lei Shi et al [16]. The posts and the recommended methods discover user interests. Once the users with great impact are identified, it is expected to predict the behavior shortly and its effect on the popularity of the events. A three-step Latent Dirichlet allocation (TS-LDA) and hypertext induced topic search based on the topic decision (TS-HITS) which could automatically unearth key posts in large amounts of posts was proposed by Lie-Lei-Shi et al [17]. Multi-layer Inverted List (MIL) [3] managed dynamic event databases during updates performed as the event evolved. But this method involves temporal decay which reduces accuracy and scalability. Vector Space Model (VSM) [24] [25] have their use in much existing research for the representation of different social events but it fails to consider information related to key posts, key users, key topics. The primary issue faced is information sparsity. Models often use a big amount of external resources to complement short text expression for topic choice. The techniques in [6] [26] rely excessively on external resources, big amounts of external resources change the original text semantics. [22] incorporated all brief texts for a particular event to a single long text document and then applied text modeling.

Nadia Derbas et al. [27] developed Safapp which was used to analyse text on social media and blogs, and on an event extractor module. The main aim of that module was to help analyse online propaganda. Safapp is a crawler to crawl data from social media and it uses Natural Language Processing (NLP) to get vital information from that data. Nadia Derbas et al. aimed on offering an app that will allow specialists a powerful app, using which they can understand the phenomenon of radicalization and online propaganda.

Aarzoo Dhiman et al. [28] proposed a model that captures contextual information using Joint Spherical Embedding (JoSE) model. To capture the relationship between Twitter data they used the weighted graph model. To detect events they employed the graph clustering model. The results obtained using the proposed method showed improved performance of up to 30% when compared to state-of-art models. Mahmud Hasan et al. [29] made use of TwitterNews+ which is an event detection system to detect both minor and major news events from real-time Twitter data. They evaluated the system using a benchmark dataset which is publicly available. The obtained results showed major improvements in the precision and recall over 5 state-of-art models.

Lida Huang et al. [30] proposed a similarity based method to detect most of the emergency events in social media, which includes accidents, natural disasters, social security and public health events. The proposed method used 3W (What, When and Where) attributes of events for clustering the text from social media. First, a 2 step classification is done to detect emergency text. Second, BiLSTM model and regular expression matching is used to extract the time and location information. Finally, clustering is performed to group social media texts into different events.

Iraklis Moutidis et al. [31] introduced a novel method to named entity disambiguation for the tweets. Then they applied sentiment analysis to the detected events to classify them into negative or positive based on how they are portrayed by the media. The proposed approach was applicable in the domains where sentiments of public were relevant to decision-making, such as politics and financial markets. Ali Momen Sani et al. [32] implemented an algorithm for real-time event detection. It used hashing as a vectorizer and to detect similar items they used locality sensitive hashing, it is then grouped with incremental clustering. They were able to detect 102 events with 87.25% precision. Han Zhou et al. [33] did an extensive survey on multi-modal social event detection. They observed that current system aims on detecting events in large heterogeneous data which can be in the form of text, image or video. They reviewed event inference and event feature learning which are recent research. They also provided a thorough comparison of various public datasets in the community.

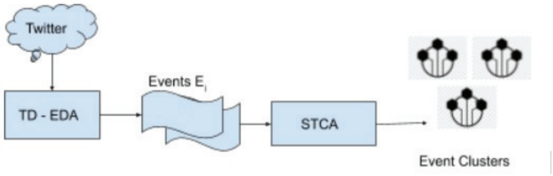

Figure 1. Event Detection Model system flow

III. Proposed Approach

The similar Event Evolution Detection (SEDM) model, refer Figure 1 shows us the SEDM model which accommodates four modules. Instinctively, first, the event detection algorithm is used to procure structured event quintuples which incorporate key traits of the events contained in the stream of tweets from Twitter API. A self-tuning clustering algorithm is proposed to combine tweets, forming clusters that are composed of key posts with similar context and hence, similar topics. The clusters formed are used for two major purposes; one, to detect similar events, and two, to discover the progression of the nature of topics contained in the event. Two separate methodologies are devised for the same. A similar event detection algorithm reveals events that overlap in both time and topics to evaluate the effects of these similar events on deliberate human actions. Topics focused on throughout an event are subject to change as the event evolves. The topic evolution algorithm puts into perspective the change in topics for an event's lifetime.

Table I
Parameters

Sr.no.	Parameter	Description
1	Common users	Key contributors in events E_i and E_j
2	Cosine similarity	Similarity of topic vectors for events E_i and E_j
3	Time divergence	Difference in $timestamps$ of events
4	Post Spread factor	Retweet count for a particular post
5	Spatial spread factor	Location of keyposts of event

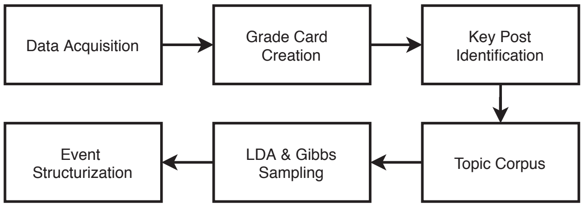

Figure 2. TD-EDA Model

A. Topic Decision based Event Detection Algorithm (TD-EDA)

In this section, we introduce a few fundamental concepts used in our paper and the research problem we aim to achieve.

Definition 1 : Event

Each event E is a structured quintuple made up of elements as follows ⟨timestamp, location, leading_post, topics, key_users⟩, where timestamp denotes the time and day when the event E was recognized, location tells us the region where the event E occurred, leading posts are an array of microblogs relevant to the detected event E, topics are a key parameter that categorizes events under various topics, key users are a list of top Z active members related to the event E.

Definition 2 : User Profile Card

In the time span T, for each user U his activeness / influence with respect to different topics is recorded and a grade card is created. Grade card for each user U denotes the degree of U's interest in topic K_i belongs to (K_1, K_2, … ,K_N). Grade card gives us the value of Influencer Credit which is the global influence value of the user on the microblogging platform. Influencer Credit is based on two parameters as follows:

1) User's Followers (uf) - This remains constant for every topic.
2) Activeness (ua) - It denotes the frequency of user's post in a specific time window.

Definition 3 : Supremacy index

Leading posts in the social media stream network are surrounded by other noisy posts. In our proposed model topic decision is based on the basis of topic-leading posts. Such posts have a higher degree of dominance as well as a higher index value. Then all the posts with higher supremacy have the highest probability of getting chosen as a different topic. The parameters that contribute in deciding the supremacy index for each post are:

1) Likes (pl) - Count of number of users who liked the post
2) Retweet (pr) - Count of number of users who retweeted the post
3) Comments (pc) - Count of number of users who commented on the post
4) Influencer Count (ic) - Count of the number of influencers with $\beta \leq 1$ associated with the post

Definition 4 : Topic diffusion

The leading key posts and the users active for those events differ in the social media microblogging. Every event is classified by leading key posts and key users. This is identified by calculating the topic-similarity distance between the two posts as the distance should be larger for different topics.

In order to achieve our goal of identifying similar events, we need to first structurize the tweets into clusters of events. For this purpose we use the Topic Decision based Event Detection Algorithm (TD-EDA model), Figure 2 shows us the TDEDA model. First, the stream of tweets from the Twitter API is retrieved for the time span from February 14th, 2019 to March 2nd, 2019 using Twitter search API... A user profile card is created for each user which consists of his id and influencer credit (β). This card specifies his global influence in the social media network. The supremacy index(δ) is calculated for each tweet in the stream which specifies its significance. The topic decision is made on the basis of two major parameters. First, one being the supremacy index (δ) amongst all the tweets, the most dominant posts are selected. Another parameter considered is the topic diffusion underlying the posts. The M rightmost upper posts can be then selected as our key posts which give us our key topics. The topics identified are then used as the "init" parameter for event classification. Among these meticulously selected topics, the posts in the microblogging network are clustered using LDA [20] and Gibbs sampling [34]. Each cluster represents a structured event quintuple of: ⟨timestamp, location, key_post, topics, key_users⟩ Refer Algorithm 1

Algorithm 1 Topic decision based Event Detection Algorithm

```
Input: Tweets retrieved using Twitter API
Assumptions:
S - List of $S_1$,$S_2$....$S_k$ posts
IU - List of users
$\beta$ - influencer_credit
$\delta$ - supremacy index
Calculation:
  1) For each user U $\epsilon$ IU
       $\beta \leftarrow$ CalculateInfluencerCredit(U)
  2) For each post P $\epsilon$ S
       Calculate $\delta$
  3) Using Affinity Matrix A (One dimension is the $\delta$ and
     the other dimension is the topic diffusion) Right upper
     corner gives us the M keyposts
  4) For each post P $\epsilon$ M keyposts
       Keywords $\leftarrow$ FindKeyword(P)
       Identify unique keywords as unique topics
  5) Apply LDA to all the posts and input the identified topics
     as well
Output: Events $E_N$ as 5-tuple structure ⟨timestamp, location,
key_post, topics, key_users⟩
```

B. Self-Tuned Clustering Algorithm (STCA)

The methodology used for clustering algorithm is described in Algorithm 2. A self tuning clustering algorithm is proposed to combine tweets, forming clusters which are composed of keyposts with similar context and hence, similar topics which can be observed in Figure 3.

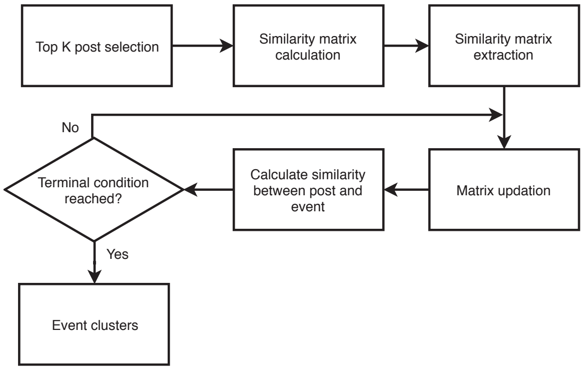
Top K post selection
Similarity matrix calculation
Similarity matrix extraction
No
Terminal condition reached?
Calculate similarity between post and event
Matrix updation
Yes
Event clusters

Figure 3. STCA Model

Algorithm 2 Self Tuned Clustering Algorithm

Input: N, the number of events; A, the link matrix; θ the maximum number of iterations.

Calculation:

```
1) Consider N posts with highest supremacy index for the first N events.
2) Create similarity matrix for any two posts.
3) Elicit the similarity matrix between posts and events. We get the initial N classes of the graph: $E_i$ after dividing the posts into the nearest event it belongs to.
4) Recalculate to update the matrices $V_{NxM}$ having weights of M post to all the N events dependent on the current partitions using step 5)
5) Compute similarity $sim_{ij}$ between post $P_i$ and event $E_j$. Form cluster of the vertices having highest similarity into N events. Thus event cluster has posts which are similar to each other.
6) Go to step 4) if terminating condition is not reached.
```

Terminating condition: No change in clustered events or the number of iterations reaches θ.

Output: Event clusters i.e all similar posts grouped into a single event cluster.

C. Similar Event Detection Algorithm (SEDM)

A Similar Event Detection Algorithm (SEDM Model) can be observed in Figure 4. Considering the number of tweets generated on the microblogging platform in consideration, Twitter, it becomes imperative to realize the effect that these tweets have on driving the views of society as a whole. Hence, it becomes necessary to not only detect the nature of events in singularity but also their nature in regards to events of a similar nature occurring in the same time frame. Keeping this essential in mind, a Similar Event Detection Algorithm (SEDM) is designed. Refer algorithm 3

From a stream of events the algorithm obtains a pair of events that are overlapping in nature. A pair of events $\langle E_i, E_j \rangle$ classify as Similar events when their topic exhibits a relation between each other in time domain. You can observed Similar Event detection Time line Chart in Figure 5.

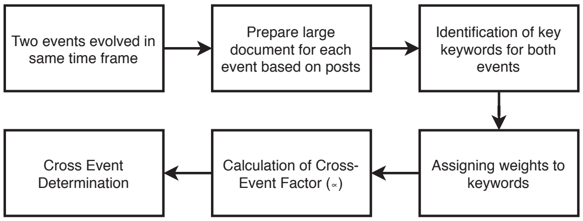
Two events evolved in same time frame
Prepare large document for each event based on posts
Identification of key keywords for both events
Assigning weights to keywords
Calculation of Cross-Event Factor (α)
Cross Event Determination

Figure 4. SEDM Model

As seen in step 1) of the algorithm, SEDM uses clustered posts to prepare large document for a particular event. This resolves the data sparsity problem of short texts in Twitter posts. In step 2), a set of key keywords is determined which helps in identifying the context of a particular event. A dictionary of keywords and counter is maintained to allow weight assignment for each keyword with respect to other keywords for the same event as seen in step 3). The above dictionary aids to establish a relation between two events on the basis of its similar or opposite nature thus, allowing calculation for similar event factor α (1) for a pair of events E_A and E_B.

$$\alpha = e^{\frac{-\delta_i}{2\sigma^2}} \tag{1}$$

where δ_i is the minimum Euclidean distance between post P_i and post P_j [16]. σ is derived from the parameters described in Table III

Algorithm 3 Similar Event Detection Algorithm

Input: Two events evolved in same time frame E_A and E_B , posts clustered in event E_A and E_B

Calculation:

```
1) $D_A \leftarrow$ PrepareLargeDocument($E_A$)
   $D_B \leftarrow$ PrepareLargeDocument($E_B$)
2) $S_A \leftarrow$ FindKeywords($D_A$)
   $S_B \leftarrow$ FindKeywords($D_B$)
3) For K $\epsilon$ $S_A$,$S_B$
     a) Assign counter for K based on occurence of word in post
     b) Make entry in dictionary for K and Counter-value
4) Calculation of similar and opposing keywords
     a) Synonyms $\leftarrow$ FindSimilarKeywords($S_A$,$S_B$)
     b) Antonyms $\leftarrow$ FindOppositeKeywords($S_A$,$S_B$)
5) $\delta \leftarrow$ Difference( len(Synonyms), len(Antonyms))
6) $\theta \leftarrow$ len(( GetTopics($E_A$) $\cap$ GetTopics($E_B$))
7) **if** $\theta \geq 1$ **then** $\alpha \leftarrow$ Combine ($\delta$, $\theta$)
   **else** $\alpha \leftarrow 0$
```

Output: Degree to which events are similar in nature

D. Topic Evolution Algorithm (TEA)

Deriving from the event detection algorithm, we obtain event quintuples containing a timestamp, location, topics, and key posts. Consequently, each event cluster is a segregated group of key posts with similar context and traits, hence revolving around a set of closely related topics in the course

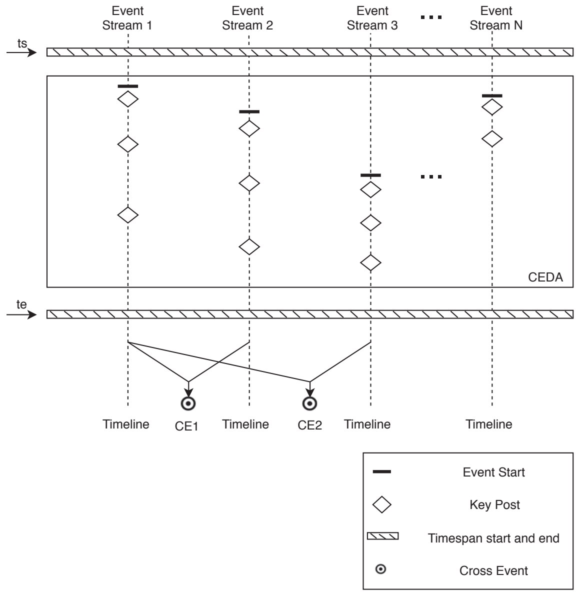

Figure 5. Similar Event Detection Timeline chart

of the lifetime of the event. The context of the focus of event topics is liable to change in the due course of the event with respect to time. It can be simply stated that if a particular topic is key at the start of the event, it may or may not continue to be key by the closure of the said event.

Taking the example of the BhimaKoregaon event, which started with topics like protests, going on to focus on bandh and reservation/ Aarakshan where the context transformed from activism to politics. In view of this, the Topic evolution algorithm (TEA) focuses on discovering the changes in the progression of corporeality contained in the event cluster. Also, we aim to predict the course of a particular event given we are aware of certain topics that have evolved in the referred event until that point in time. Refer Algorithms 4 and 5

```
Algorithm 4 Topic Evolution Algorithm : Phase 1
Calculation: For each topic in cluster E,
  1) Sort keyposts by time
  2) Split keyposts according to time window $T_w$
       a) keyword ← FindKeyword(S)
       b) AppendToList(KeywordList, keyword)
  keyword$_{i+1}\rangle$ $\epsilon$ KeywordList,
       a) similarityIndex ← CosineSimilarity(keyword$_i$, keyword$_{i+1}$)
       b) AppendToList(SimilarityList, similarityIndex)
Output: SimilarityList with cosine similarity of each pair of keywords
```

$$cos(\theta) = \frac{A.B}{\|A\|.\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}} \tag{2}$$

where A_i and B_i are components of vector A and B respectively.

```
Algorithm 5 Topic Evolution Algorithm : Phase 2
Input: An input matrix indicating topics contained in a the event
Calculation: For each topic in cluster E,
  1) All events are weighted with respect to similarity with current event. Similarity measure used is adjusted cosine similarity
  2) $N$ active events with higher similarity are selected
  3) Compute prediction P$_{a,b}$ as similarity between two events using
       $P_{a,b} = \frac{\sum_{i=1}^{rn}(r_{a,i} - \overline{r}_a) \times (r_{b,i} - \overline{r}_b)}{\sqrt{\sum_{i=1}^{rn}(r_{a,i} - \overline{r}_a)^3 \times \sum_{i=1}^{rn}(r_{b,i} - \overline{r}_b)^3}}$   (3)
     where $r$ is actual matrix value and $\overline{r}$ is the mean value
  4) Compute prediction for event $a$ on topic $t$ as P$_{a,t}$
       $P_{a,t} = \overline{r}_a + \frac{\sum_{i=1}^{n}(r_{i,t} - \overline{r}_i) \times P_{a,i}}{\sum_{i=1}^{n} P_{a,i}}$   (4)
     where $n$ is the number of events in the neighbourhood.
Output: Association rule matrix for various topics
```

In the next phase of the TEA Algorithm 5, we use the concept of collaborative filtering to predict potential future topics that may arise for the current event based on the history of events and their topics. The algorithm uses a matrix consisting of entries of events against their topics. It calculates prediction values for events to determine trailing topics.

IV. Experimentation Results

In this section, we communique the results of an extensive study conducted on a vast real-life tweet dataset. As we know, our proposed SEDM model can detect various events and identify similar events amongst them. The experiments are designed to evaluate the accuracy and efficiency of SEDM. The results show events detected, identified similar Events, and the effectiveness in mapping evolution of the various detected Similar Events.

And the rest of the section talks about, our gathered collection of data, experimental setup and analysis, the baseline approach, and the model evaluation.

A. Dataset

We generated our dataset from Twitter [12] via Twitter API. This dataset consists of 2,78,817 tweets posted from February 15th, 2019 to March 2nd, 2019 using Twitter search API. We use the Search API to retrieve “meaningful” tweets by giving a list of search terms consisting of event-related words (example, rape, accident, terrorism). Every tweet in our dataset undergoes preprocessing which removes stop words, stemming, and obtains nouns and verbs only. Further, we consider only those tweets which are unique (do not count the tweet which is just a retweet). Again for the users, we count the ones who have either posted the tweet or commented on the tweet. We get a total of 45,948 high-quality tweets.

B. Setup

The experimental setup was rendered on a machine with Intel I5 7th generation processor and NVidia GeForce GTX 1050 2GB GPU. Required parameters were tuned. For PLSA, LDA and HEE, the blend weight of the background model B were fixed to 0.05 [8]. For LDA, = 0.5 and = 0.1. In all the strategies, Gibbs inspecting was kept running for 1,000 iterations. EM Algorithm was additionally kept running for 1,000 cycles. The results reported are the normal more than 5 runs. During the time spent filtering high-quality posts and high-impact users, the majority of the initial authority scores were set to 1

C. BASELINE APPROACHES

The productivity and adequacy of the proposed model is approved by assessing our model against PLSA, LDA, and HEE [7], which are the classic latent semantic analysis algorithms.

D. Evaluation Phases

For our experiments, we employed the dataset containing microblogs on the recent event that occurred in India,i.e. The Pulwama Attack. The events present on the Twitter platform are highly heterogeneous. Our work is to first cluster all the events together, find the similarity between them, and keep track of how they evolve in the future. This information then helps us to predict event evolution for another topic under the same category.

1) Topic Decision based Event Detection Algorithm: First, we create User Grade Card to find user's global influence on Twitter platform. Each Grade Card consists of user's ID and Influencer Credit value β. The value of β is calculated on the basis of user's followers and activeness along with the count of retweet. Table II shows the Grade Cards of the Key Users.

Table II
USER PROFILE GRADE CARD

User ID	Grade Card Value β
UserID : 92724677	0.0561
UserID : 101311381	0.0287
UserID : 4191157280	1.032
UserID : 4887789076	2.031
UserID : 375495587	0.297
UserID : 35817960	0.0008
UserID : 2917237769	1.440

Then we identify the key posts by calculating its supremacy index value δ. Table 2 gives us the key posts along with its key topic identification.

Table III explains the main users similar event posts with regard to all changing activities in the same time frame, which further increases the effectiveness of event detection.

Major events are determined on the basis of supremacy index. The key posts having a huge difference between its value of supremacy index are considered to be two different events.We list the events in the appropriate structured format as shown in Table IV. After identifying the key terms in all the tweets and major events, clustering can be performed.

Table IV
STRUCTURIZED EVENT QUINTUPLES

Event	Event Quintuples
E1	< 27th feb 2019, BringBackAbhinandan, [Abhinadan, SurgicalStrike, WingCommandar] >
E2	< 26th feb 2019, BalakotAirStrike, [Balakot, AirStike, Mig21, SurgicalStrike] >
E3	< 25th feb 2019, BloodDonationbyDSS,[BloodDonnation, Matryed, Camps, DSS] >
E4	< 14th feb 2019, Pulwama, [Pulwama, Attack, JEM, Kashmir, CRPF] >
E5	< 27th feb 2019, SayNoToWar, [saynotowar, peace, india, pakistan] >

2) Self Tuned Clustering Algorithm:

a) Doc2Vec: It is an extension of word2vec approach, which adds additional input parameter, document id. Here we gather all the tweets together and prepare a document of all microblogs. This documents goes through the Doc2vec model to create vectors. We cluster the Doc2vec vectors using k-means algorithm.We use the average silhoutte method to decide the value of k. The value k=5 gives the best silhoutte score as shown in the Table V. Fig 6 shows the density of texts in each of these 5 clusters. Table VII gives us the top terms in each of the 5 clus-ters formed

Table V
DETERMINING VALUE OF K

Value of k	score
2	0.5126866
3	0.12053437
4	0.09570506
5	0.13136749
6	0.056070995
7	0.05263376
8	0.053256914
9	0.05496637
10	0.05444773

Table III
KEY POST IDENTIFICATION

Username	Tweet	Retweet Count	Likes	Supremacy Index δ	Timestamp	Key Topic
'Virender Sehwag'	'How proud we are to have you ! Bow down to your skills and even more your grit and courage #WelcomeBackAbhinandan'	11015	90313	0.404	'27 Feb 2019'	'Abhinandan'
'Shah Rukh Khan'	'There is no better feeling than Coming back Home, for home is the place of love, hope and dreams. Ur bravery makes us'	12342	90553	0.182	'1st March 2019'	'Abhinandan'
'Ashesh Pattnaik'	'Attack on CRPF personnel in Pulwama is despicable. I strongly condemn this dastardly attack. The sacrifices of our brave'	41371	23450	0.447	'15th Feb 2019'	'Pulwama'
'Imaandar KN Tripathi'	'We have entered balakot to kill azhar masood who is criminal for pulwama attack. Now without Azhar masood our opera'	1	7	1.782	'27th Feb 2019'	'Balakot'
'Hans Masroor Badvi'	'How India Treated Sipahi Maqbool HussainHow we are treating AbhiNandan IS TERRORIST?#SayNoToWar'	680	1251	0.839	'27th Feb 2019'	'Say NoTo War'
'Archana'	'As a Tribute to Pulwama Martyrs BloodDonationByDSS will be organized on 27 feb 2019 worldwide.'	107	26	1.090	'25th Feb 2019'	'Blood Donation'

Table VI
TOP TERMS PER CLUSTER USING DOC2VEC

Cluster Number	Event	Key Terms
0	E1	' abhinandan war pulwama welcomebackabhinandan bringbackabhinandan peace pakistan india abhinandanreturns want welcomehomeabhinandan pulwama pulwamaattack martyrs indian pakistan tribute'
1	E2	' pulwama blooddonationbydss tribute martyrs pakistan india abhinandanreturns war blood attack and pulwamaattack people'
2	E3	' pulwama tribute martyrs pulwamaattack ,kashmirterrorattack pulwamaterrorattack crpf '
3	E4	'saynotowar blooddonationbydss pulwama pulwama martyrs tribute blood feb donation 27th camps organising abhinandan war attack india abhinandanreturns'
4	E5	'abhinandan welcomebackabhinandan back pulwama abhinandanreturns india peace welcomehomeabhinandan bringbackabhinandan pakistan i want welcome salute war hero'

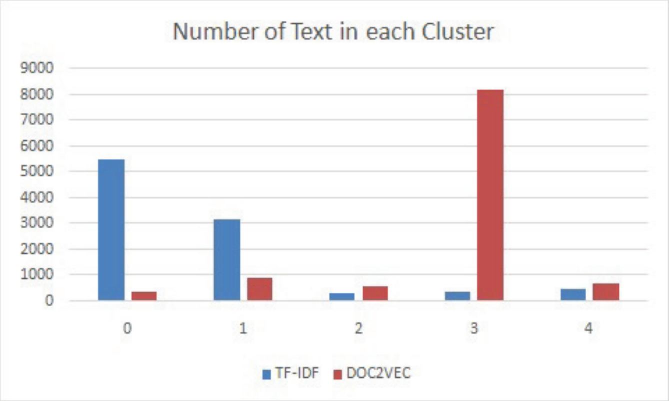

Figure 6. Doc2Vec and TF-IDF Methodology

b) Term Frequency-Inverse Document Frequency: TFIDF is Term Frequency-Inverse Document Frequency. The term frequency of a word W is defined as the number-time ratio of word W appearing in a document to the total number of words in the document. Inverse Document Frequency is provided as the log of the proportion of the complete amount of papers to the number of documents with word W. TF-IDF is a simple multiplication of TF and IDF. This approach works on the principle that relevant words are not necessarily the most frequent words. TF-IDF helps us to find the relevant words in the document and cluster them. For clustering using a weight matrix, we calculate the cosine similarity between two documents. Figure 6 shows how tweets are grouped into 5 clusters.

Table VII gives us the top terms in each of the 5 clusters formed. These clusters are belonging to events related to the Pulwama attack. so these clusters are events cluster 0 represents Abhinandan Event (E1), cluster 1 represents BalakotAirStrike Event (E2), cluster 2 represents BloodDonation Event (E3),cluster 3 represents PulwamaAttack Event (E4), cluster 4 represents SayNoToWar Event (E5).

c) Latent Dirichlet Allocation: LDA views each text as a mixture of topics and each topic as a mixture of words, all of which can be overlapping. It gives several word vectors consisting of similar key terms. LDA groups tweets into 10 topics.

Precision and efciency of Event-detection: To compare the precision and efficiency of our model with those of the PLSA, LDA, and EVE models, we evaluated effectiveness in our experiments. Table For PLSA, LDA, HEE, and our SEDM model, we denied precision as follows:

$$Precision = \frac{The\ number\ of\ detected\ events\ matching\ real\ life\ events}{The\ Total\ number\ of\ the\ events\ detected}. \quad (5)$$

Table VIII shows the precision of all the methods. Our methods outstand among all methods.

Table VII
TOP TERMS PER CLUSTER USING TF-IDF

Cluster Number	Event	Key Terms
0	E1	'Welcomebackabhinandan, gobackmodi, abhinandanmyhero, jaihind, abhinandancomingback, hero, sir, proud, salute, pakpremipatrakaar, abhinandandiwas, abhinandanvarthaman, thankyouimrankhan, braveheart, abhinandanreturns, whereismugilan, saynotowar, wait, pakistanleadswithpeace'
1	E2	'bringbackabhinandan, saynototwar, far, updates, lives, abhinandan, war, please, india, peace, pakistan, nationfirst, bringbavkabhi, safe, mig21, balakotairstrike, bring'
2	E3	'Blood, donating, camps, blooddonationbydss, tribute, known, humanitarian, standing, organised, pulwama, condict, martyred, 27th, sacha, feb, martyred, singh, saints, volunteers'
3	E4	'Attack, pulwama, crpf, terrorist, india, condemn, security, pakistan, kashmir, pm, soldiers, modi, indian, jawans, separatists, celebrating, families, news,withdraws, nation, cover, kashmireterrorattack, trending, need'
4	E5	'saynotowar, heartfelt, heartbroken, hear, headline,hate, hero, highly,hats'

Table VIII
COMPARISON OF PRECISION

Method	K=1	K=5	K=8	K=10
PLSA	1/1	5/5	-	-
LDA	1/1	5/5	-	10/10
HEE	1/1	5/5	6/8	-
SEDM	1/1	5/5	8/8	10/10

Table IX
COMPARISON OF TIME

Method	HITS	Topic Choice Based	Gibbs sampling	EM	Total
PLSA	0	0	0	36.78min	36.78min
LDA	0	0	24.68min	0	24.68min
HEE	29897ms	6.89min	4.05min	0	11.43min
SEDM	28889ms	5.98,min	3.58min	0	10.04min

Table IX shows comparative results of PLSA, LDA, HEE models, and SEDM models. As per the numbers given in the table if K is set for 8, our proposed SEDM model can find seven events. However, if K is set any greater than 6, then the PLSA, LDA, and HEE models can sense all events only by artificial selection. All events would stay undetected if K is set to 1 or 5. Time efficiencies of all these models are very less if the K value is greater than 8, such as 10. Our SEDM model is therefore both precise and efficient, and its efficacy is better than that of the PLSA, LDA, or HEE models.

3) similar Event Detection Algorithm:

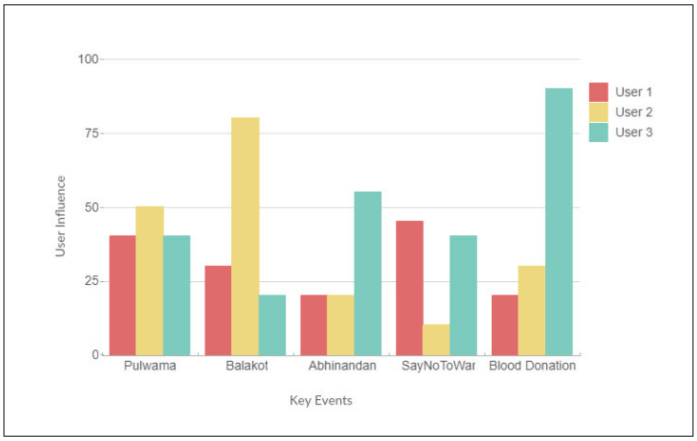

Figure 7. similar Event Analysis : User Based

a) User Based Analysis: : It describes how users may react differently to two different events under the same category. The graph below shows how an individual's opinion varies as the context under the events of the same category change. Figure 7 gives us an analysis of how users influence vary as the event goes on evolving.

b) similar Topic Based: It defines the wisdom of similar events key posts, Key topics, their key users, and their weight event wise. It demonstrates the main subjects of significant Pulwama associated activities and how the weights of the subject are altered with the evolution of events. The graph above shows how similar-event topic changes. Figure 8 provides us the impact of similar-event topics assessment as the event continues to evolve.

c) similar Event Detection Matrix: It defines the degree to which all events are similar in nature. Using SEDM algorithm the factor of similar incidents is calculated. Table X demonstrates us how events are similar in nature. It demonstrates that all occurrences from E1 to E5 are all similar in nature because their similar event factor is higher than 0.5.

Table X
SIMILAR EVENT MATRIX

similar Event Matrix	E1	E2	E3	E4	E5
E1	0				
E2	0.55	0			
E3	0.53	0.52	0		
E4	0.45	0.52	0.51	0	
E5	0.53	0.53	0.51	0.51	0

4) Topic Evolution Algorithm: Topic evolution algorithm is concerned with pattern of evolution of topics within clustered events. With respect to algorithm 4, cosine similarity index is used to discover top keywords and their weights as event progresses in time. Tweets with top keywords are then extracted from the clustered events to predict new relationships among the keywords with the method described in algorithm 5.

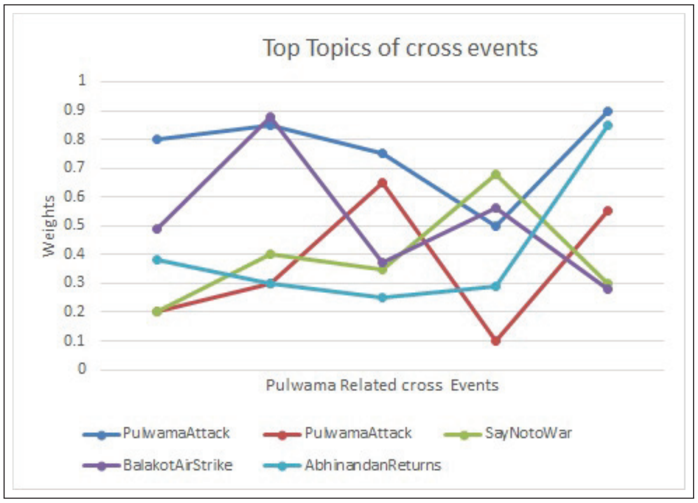

Figure 8. Events weight timeline evolution

The results from Topic evolution algorithm can be observed in table XI.

Table XI
SET OF NOTABLE RULES OBTAINED FROM TEA

Sr.no.	Rule	Lift
1	'blood', 'volunteers' → '27th'	51.98
2	'camps', '27th' → 'volunteers'	59.15
3	'air', 'strike' → 'terror'	25.42
4	'heartfelt', 'martyred' → 'attack'	84.00
5	'pulwama', 'attack' → 'donating', 'families'	106.70
6	'killed', 'want' → 'peace'	22.83
7	'violation', 'ceasefire' → 'pakistan'	62.87

V. CONCLUSION

Social media is widely used to share information globally and it also aids to gain attention from the world. However, Discussions in state-of-the-art papers are restricted to event detection, user interest discovery, and detecting influential spreaders but overlook similar event detection and topic evolution concerning similar events. This leads to difficulty in tracing the nature of evolving events. To address these problems, we propose the SEDM Model. It consists of two main algorithms Similar Event detection Algorithm (SEDM) and Topic Evolution Algorithm (TEA). Similar Event Detection is critical in determining similar events, which have many fulcrum points, so the SEDM algorithm detects similar events that are similar with regards to their temporal nature resulting from main events. The topic evolution algorithm is concerned with the pattern of evolution of topics within clustered events. For the algorithm, the cosine similarity index is used to discover top keywords and their weights as the event progresses in time. The topic evolution algorithm is concerned with the pattern of evolution of topics within clustered events. Moreover, this approach can further find topics that are shaping the behavior of the event. Data sparsity issue is also rectified by appending all tweets in one document for all similar events. Finally, the experimental results on a real Twitter dataset demonstrate the efficiency and accuracy of our proposed model for both similar event detection and topic evolution.

Further, the model finds similar kinds of events and users related to those events. This helps to identify similar events. It allows us to understand the behavior of users towards various evolving similar events. We also learn about how an individual's opinion changes as the events evolve into another event even though it falls under the same context. As we discover how events grow and their evolution pattern, we can determine the relationship between various events that may or may not be similar. This helps us to even predict what could be the next event that may arise after the occurrence of such similar event trails.

In future work, to understand the impact and evolution of growing events on a larger radius of society, we plan to further predict the behaviors of the user community based on the dynamic community detection model. To improve the efficiency of our model we can integrate other social media data e.g. Instagram check-in data for accurate predictions.

REFERENCES

[1] X. H. Amber Umair, Priyadarsi Nanda, "Online social network information forensics," *IEEE Trustcom/BigDataSE/ICESS*, 2017.

[2] S. S. Wen-Yu Lee, Winston W. Hsu, "Learning from cross-domain media streams for event-of-interest discovery," *IEEE Transactions on Multimedia, VOL. 20*, January, 2018.

[3] D. S. Hongyun Cai, Zi Huang and Q. Zhang, "Indexing evolving events from tweet streams," *IEEE Transactions on Knowledge and Data engineering, vol. 27, no. 11*, November, 2015.

[4] B.-C. X. Chung-Hong Lee, Hsin-Chang Yang, "Exploring cross-event relations on twitter datasets via topic recommendation and word embedding," *IEEE 8th International Conference on Awareness Science and Technology*, 2017.

[5] J. L. Abdulrahman Aldhaheri, "Event detection on large social media using temporal analysis," *IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, 2017.

[6] G. D. T.-S. C. Sicheng Zhao, Yue Gao, "Real-time multimedia social event detection in microblog," *Published in: IEEE Transactions on Cybernetics (Volume: 48 , Issue: 11)*, Nov. 2018.

[7] I. C. F. A. G.-J. H. Elena Ilina, Claudia Hauff, "Social event detection on twitter," *ICWE International Conference on Web Engineering*, 2012.

[8] C. Li, A. Sun, and A. Datta, "Twevent: Segment-based event detection from tweets," *ACM*, 2012.

[9] P. K. Nikolaos D. Doulamis, Anastasios D. Doulamis and E. M. Varvarigos, "Event detection in twitter microblogging," *IEEE Transactions on Cybernetics, vol. 46*, December, 2016.

[10] Q. L. B. G. Zhiwen Yu, Fei Yi, "Identifying on-site users for social events: Mobility, content, and social relationship," *IEEE Transactions on Mobile Computing*, 2018.

[11] R. Z. W. Y. L. L. Jianxin Li, Zhenying Tai, "Online bursty event detection from microblog," *IEEE/ACM 7th International Conference on Utility and Cloud Computing*, 2014.

[12] "Twitter," http://www.twitter.com.

[13] M. I. P. M. Dat T. Nguyen, Ferda Ofli, "Damage assessment from social media imagery data during disasters," *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2017.

[14] S. Luna and M. Pennock, "Social media in emergency management advances, challenges and future directions," *Annual IEEE Systems Conference (SysCon) Proceedings*, 2015.

[15] C.-L. Hu, "On-demand real-time information dissemination: A general approach with fairness, productivity and urgency," *International Conference on Advanced Information Networking and Applications*, 2007.

[16] Y. W. L. J. J. H. Lei-Lei Shi, Lu Liu, "Event detection and user interest discovering in social media data streams," *IEEE Access, vol. 5*, 2017.

[17] L. L. X. S. Leilei Shi, Yan Wu and L. Jiang, "Event detection and identification of influential spreaders in social media data streams," *IEEE Big data mining and analytics*, March 2018.

[18] S. X. XIAOHUI ZHAO, FANG'AI LIU and Q. WANG, "Identifying influential spreaders in social networks via normalized local structure attributes," *IEEE Access*, 2018.

[19] S. W. . S. O. Silvia Planella Conrado, Karen Neville, "Managing social media uncertainty to support the decision making process during emergencies," *Journal of Decision Systems*, 2016.

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

[21] T. Hofmann, "Probabilistic latent semantic indexing," *Proc. 22nd Annual Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., vol. 8., pp. 50-57*, Aug. 1999.

[22] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie, "Improving lda topic models for microblogs via tweet pooling and automatic labeling," *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM*, 2013.

[23] I. O. P. H. Anjie Fang, Craig Macdonald, "Topics in tweets: A user study of topic coherence metrics for twitter data," *Advances in Information Retrieval. ECIR 2016. Lecture Notes in Computer Science, vol 9626. Springer, Cham*, 2016.

[24] L. R. Y. E. Olga Peled, Michael Fire, "Entity matching in online social networks," *International Conference on Social Computing*, 2013.

[25] C. Musto, "Enhanced vector space models for content-based recommender systems," *Proceedings of the Fourth ACM Conference on Recommender Systems*, 2010.

[26] Z. Chen, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, "Discovering coherent topics using general knowledge," *Proceedings of the 22Nd ACM International Conference on Information and Knowledge Management*, 2013.

[27] N. Derbas, E. Dusserre, M. Padró, and F. Segond, "Eventfully safapp: hybrid approach to event detection for social media mining," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 1, pp. 87–95, 2020.

[28] A. Dhiman and D. Toshniwal, "An approximate model for event detection from twitter data," *IEEE Access*, vol. 8, pp. 122 168–122 184, 2020.

[29] M. Hasan, M. A. Orgun, and R. Schwitter, "Real-time event detection from the twitter data stream using the twitternews+ framework," *Information Processing & Management*, vol. 56, no. 3, pp. 1146–1165, 2019.

[30] L. Huang, G. Liu, T. Chen, H. Yuan, P. Shi, and Y. Miao, "Similarity-based emergency event detection in social media," *Journal of Safety Science and Resilience*, 2020.

[31] I. Moutidis and H. T. Williams, "Good and bad events: combining network-based event detection with sentiment analysis," *Social Network Analysis and Mining*, vol. 10, no. 1, pp. 1–12, 2020.

[32] A. M. Sani and A. Moeini, "Real-time event detection in twitter: A case study," in *2020 6th International Conference on Web Research (ICWR)*. IEEE, 2020, pp. 48–51.

[33] H. Zhou, H. Yin, H. Zheng, and Y. Li, "A survey on multi-modal social event detection," *Knowledge-Based Systems*, p. 105695, 2020.

[34] R. Alhamzawi and K. Yu, "Variable selection in quantile regression via gibbs sampling," *Journal of Applied Statistics*, vol. 39, no. 4, pp. 799–813, 2012.