Hindawi
Complexity
Volume 2021, Article ID 8833084, 15 pages
https://doi.org/10.1155/2021/8833084

Review Article

Topic Detection and Tracking Techniques on Twitter: A Systematic Review

Meysam Asgari-Chenaghlu,[1] **Mohammad-Reza Feizi-Derakhshi**,[2] **Leili Farzinvash**,[3] **Mohammad-Ali Balafar**,[3] **and Cina Motamed**[4]

[1]*Department of Computer Engineering, University of Tabriz, Tabriz, Iran*
[2]*Computerized Intelligence Systems Laboratory, Department of Computer Engineering, University of Tabriz, Tabriz, Iran*
[3]*Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran*
[4]*Department of Computer Science, University of Orléans, Orléans, France*

Correspondence should be addressed to Mohammad-Reza Feizi-Derakhshi; mfeizi@tabrizu.ac.ir

Received 14 February 2021; Revised 9 May 2021; Accepted 8 June 2021; Published 18 June 2021

Academic Editor: Fei Xiong

Social networks are real-time platforms formed by users involving conversations and interactions. This phenomenon of the new information era results in a very huge amount of data in different forms and modalities such as text, images, videos, and voice. The data with such characteristics are also known as big data with 5-V properties and in some cases are also referred to as social big data. To find useful information from such valuable data, many researchers tried to address different aspects of it for different modalities. In the case of text, NLP researchers conducted many research studies and scientific works to extract valuable information such as topics. Many enlightening works on different platforms of social media, like Twitter, tried to address the problem of finding important topics from different aspects and utilized it to propose solutions for diverse use cases. The importance of Twitter in this scope lies in its content and the behavior of its users. For example, it is also known as first-hand news reporting social media which has been a news reporting and informing platform even for political influencers or catastrophic news reporting. In this review article, we cover more than 50 research articles in the scope of topic detection from Twitter. We also address deep learning-based methods.

1. Introduction

Topic detection and tracking, which is also called TDT, is techniques and methods used for detecting news or document related topics best fitting their relevant intellectual material and also tracking these events or detected topics through dedicated media. Topic detection is a summarization problem that must fulfill certain demands. Topic as a summarized tag-set of an input document is different from an event which in most cases is a real-world phenomenon with certain spatial and temporal properties [1, 2]. This tiny difference between a topic and an event becomes more clear when talking about social networks. Identification of ongoing events on media can be expressed as *detection* while tracking of these events and storyboarding is *tracking*. This so called media can be a single document, group of multiple documents, or even a social media like Twitter. Topic detection and tracking has been widely applied to documents, offline corpus, and newswire, including a pilot study running from 1996 till 1997 and sponsored by DARPA [3].

Social media services like *Twitter*, *Facebook*, *Google+*, and *LinkedIn* play an important role in information exchange. In case of *Twitter*, the data exchange metrics predict that 7,454 tweets are sent per second which are about 644,025,600 tweets per day [4]. This metric for 2013 was reported by Twitter officials to be more than 500,000,000 per day [5]. Importance of this large amount of data that has large variety of topics which users tend to talk about comes to light when researchers revealed that users are most likely to talk about real-world events in social media networks more than traditional *news* and *blogging* media. Detection of topics on these short messages can make a more describing insight of users opinions about named events and real-world occurrences.

A new research area of this TDT race has begun while new social media like *Twitter* has come to existence. *Twitter* by its nature is composed of users instantly sending short posts called *tweets*. These *tweets* can be daily life messages of a user such as "*i ate a pizza! yaaay!*"; important messages from a technical society like "*Ubuntu 16.10 release date is soon!*"; or even a political message like "*WikiLeaks operative: Clinton campaign emails came from inside leaks, not Russian hackers.*" These messages are often tagged with specific word to make it addressable and fetchable. Figure 1 shows an example of tagging in *Twitter*. However, mostly this tag does not show much relation between desired news and topics, only a user's point of view in relation to his/her *tweet*. One message can be about voting while another is related to feeding ducks and both are tagged as *#DuckTales*. This issue can be addressed as *variety* from big data aspect and *ambiguity* from natural language processing aspect. Moreover, detection of a real-world event with large *volume* and *velocity* of data requires more research than finding an event on selected and filtered datasets [6]. Another problem with this media is noisiness of posted tweets. These tweets, unlike news articles and intellectual documents, are not well written and contain misspelling, grammatical errors, and even words or expressions like "*yaaaaaay*" that are not literary. Expressed problems of this media make TDT task much harder.

Data mining and artificial intelligence community has seen many research works done in this scope which show promising leverage compared to each other. Many of these works are based upon simple *bag of words* model while others keep searching on *probabilistic topic models* and still some of them look for sudden change in monitored properties. The common part of them all is the use of natural language processing techniques and methods instead of character level stochastic *n*-gram models.

These methodologies have come to aid in accomplishing the task of detecting and tracking events, and topics on social media streamlines are emerging to answer couple of questions such as the following:

(i) What everybody talks about in a specific time?

(ii) What is trending?

(iii) What happens somewhere on Earth?

(iv) Also, dynamic answered questions which have temporal and spatial properties with great increase of public interest.

In order to find most related articles to this scope, we used Google Scholar academic search engine. First we prepared our search keywords that are listed as follows:

(i) Topic detection

(ii) Twitter topic detection

(iii) Twitter event detection

(iv) Twitter event extraction

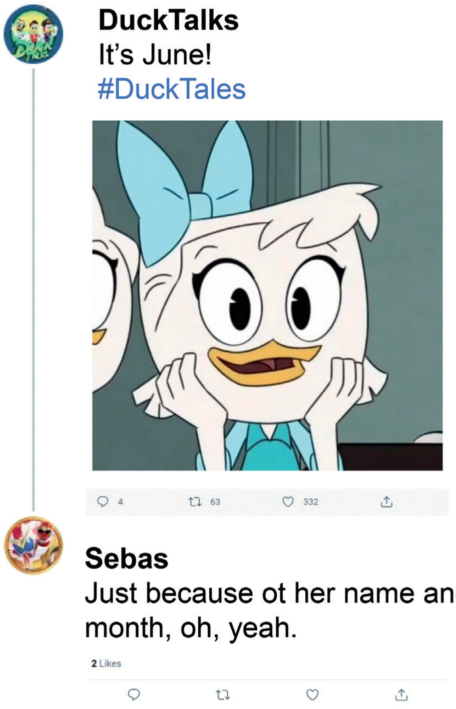
(v) Twitter topic extraction

(vi) Twitter topic tracking

(vii) Twitter event tracking.

(viii) Twitter trending topic.

(ix) Twitter trending event



DuckTalks
It's June!
#DuckTales

Sebas
Just because ot her name and this month, oh, yeah.

2 Likes

Figure 1: Some *tweets* related to a hashtag: *#DuckTales*.

We used citation per year metric to get an overall metric of importance of each article from an academic point of view. We used a threshold of two for this metric and eliminated articles that had less than two citations per year. In case of new articles, such as the ones that are published in the past two years, we did not remove them from the list even if they have less than two citations per year. In order to make sure that the unrelated articles are eliminated from the list, we read the title and abstract of each article and eliminated ones that are not related to our review title. Afterwards, we categorized the remaining articles based on their novelty and methodology. The remaining articles are the ones used to conduct this research.

This review article is organized as follows: Firstly, Section 2 describes *Twitter* as a service. Section 3 categorizes and explains existing methods and models. In Section 4, preprocessing as a general step which is common between methods is explained. Section 5 details the methods and approaches based on different categorizations. Section 6 provides a general discussion about data and evaluation issues. At the end, Section 7 concludes the paper.

2. Twitter

Twitter is described in the current section and its respective features are detailed. In Section 2.1, this microblogging service and its data types are explained. Section 2.2 discusses the details of TDT task obstacles in case of Twitter. Finally, in Section 2.3, the social big data tools are explained and detailed.

2.1. Twitter Microblogging Service. Twitter as one of the largest social blogging services is the world's fifteenth website, and in the United States of America it is the ninth and has been linked by over 6,087,240 websites (extracted from Alexa website). Its services include posting of short text messages on online Twitter platform which also enables users to track posted short messages of other users by *following* them. These short messages are called tweets that may contain a GIF image, a short textual message containing 140 characters or less including some *emojis* or only text, an image, or a poll. All of these parts are listed as parts of a *tweet*:

(1) A short text message composed of 140 characters or less that can contain emojis

(2) An image

(3) A GIF describing short text message, feeling, or anything else

(4) A poll question with predefined answers (only one of last three parts of a *tweet* can be used)

Twitter allows users to communicate in their respective social network with other users by these tweets. They can share their ideas, feelings, polling questions, pictures, and anything else that has no contradiction with its rules. A tweet posted on Twitter can be seen by other users by default unless users change their privacy settings to make it readable to only followers list or specific people.

A *Mention* or *Reply* tweet can be made by using "@" symbol before a user name. These replies or mentions create a more social web service by helping users to interact with and reply to each other. Retweet is also another feature of Twitter that allows users to resend or forward another user's tweet to their respective followers. Hashtag is also another feature of Twitter that helps users categorize their tweets with use of a "#" sign and a word related to the posted tweet; this simple keyword style helps in tweets retrieval and categorization and is also used by Twitter to detect trending events.

Twitter also provides an application programming interface (Twitter API) that enables developers and researchers to access its streaming tweets. This streaming can be filtered out by location, specific keyword, author, etc.

2.2. Challenges of Twitter for Event Detection and Tracking Task. Twitter as a great information source that is described in the earlier section has enormous information retrieval issues that make event detection and tracking task in its growing social network much harder. Twitter streams usually contain large amount of rumor tweets that have been generated by users or spammers. These fable, fiction, and in most cases mendacity tweets greatly affect the performance of event detector and tracker systems. Another issue arises when most of tweets are related to daily life of users, that is, about their personal information and daily activities. In some cases such as elections, these daily activities can be used to retrieve good information, but in the case of general event detection, they are not so much helpful. For a good event detector and tracker system, it is necessary to separate this irregular and polluted information from useful information.

Twitter messages are as short as 140 characters as the maximum size, which raises another problem. These short messages must be grouped or preprocessed to make a longer stream of tweets. Event detection and tracking in general long documents and newswire is much easier in terms of sparsity and irrelevance of documents than in the case of short blogging services such as Twitter. Most of Twitter posts contain grammatical errors and misspellings that make it harder compared to regular newswire. Twitter, as a source of user generated data, mostly contains many unseen words that are only seen in short messages. As an example of such words and abbreviations, we can name the word "OMG" which is equivalent to "Oh My God"; such words are used and generated frequently by users. Users also add misspelling and lengthen to such words, which results in a very unpleasant issue.

All of the mentioned problems are also added to big data 3-V model in which a large *variety* of *velocity* data along with big *volume* are generated and need to be processed just-in-time to be monitored and tracked. This 3-V model is much more generalized than the 5-V model that is defined as follows:

(i) *Volume* denotes large amount in terms of tally about data that is streamed or generated. Processing, grouping, clustering, and making useful information out of large scale data are crucial in information retrieval applications and also in case of Twitter-like social networks.

(ii) *Velocity* indicates speed of data generation or transfer. Streaming and online data sources such as Twitter possess this property in which real-time information extraction applications are needed to fit this kind of speed.

(iii) *Variety* is called difference of data gathered from a data source in which various data types are generated and collected to be processed. In case of Twitter, this data is different because users' generated data types are about distinct topics and events.

(iv) *Value* describes the process of information extraction from big data sources. It is also known as big data analysis that in case of Twitter is noted as *big social data analysis.*

(v) *Veracity* refers to correctness and accuracy of information extracted from a big data source. It is also known as data quality [7]. This quality is poor for some tweets (user generated daily life tweets) while it is rich in case of Twitter newswire accounts (such as a news channel related Twitter account that only posts rich tweets about real-world events).

2.3. Social Big Data Tools. Many tools for different applications of social big data analysis, storage, database systems, cluster computing, web crawler, data integration, parallel data flow, and complex event processing are presented by different companies. These tools are trivial for today's big data analysis and of course for Twitter data analysis. Some methodologies in this review use some of these tools while others do not:

(1) *Lucene* is a free and open-source information retrieval java library that has been ported to other programming languages such as PHP, C#, C++, Python, and Ruby. Indexing, searching, and recommendation are other capabilities of this tool. It has its own mini-query syntax which is easy to grasp, and its nature helps researchers and information retrieval industry to use it as a free and open-source Apache foundation tool [8, 9].

(2) *Apache Storm* is another free and open-source real-time computing system. It can reliably process unbounded streams of data for real-time applications. It is simple and can be used with any programming language [10].

(3) *NoSQL databases* such as MongoDB are designed to store and retrieve any data with big data properties in large scales. Social data storage and retrieval require NoSQL databases to perform computing tasks [11].

Other tools and programming languages can be used in this particular job, but the main properties of social big data require making use of the described tools as their relativity.

3. Categorization of Methods

Existing methods for event detection and tracking task in Twitter can be categorized in different ways based on diverse points of view. One of these categorizations distinguishes between methods that *only detect* versus methods that *detect and track* events. Some of existing methods only detect while others track detected events and make storyline of detected topics based on timeline of tweets. The first one is also be known as *topic detector* while the other one realizing importance of tracking is an *event detector and tracker*, respectively, abbreviated as *TD* and *EDT*.

Another categorization is raised when different methods use different Twitter data sources. Some use offline datasets for detection and/or tracking while others make use of online Twitter API. This distinction of data acquisition for training and testing part of algorithms raises a comparison error when comparing performance and results accuracy of existing methods.

Two other categories for event detection and tracking are known as retrospective event detection and new event detection. These two are abbreviated as *RED* and *NED*. The main focus of RED is to discover previously unidentified events from offline datasets and documents while NED is focused on finding new events in online data streams. For TDT tasks, these two concepts are broadly investigated, and many research articles have been published to fulfill this task. From Twitter point of view, event discovery algorithm can be either NED or RED. Iterative clustering algorithms such as k-means are a common practice in RED category. Firstly, a document, sentence, or short tweet is selected as an entity and other entities are compared to the first one; if it is close enough in terms of distance in vector space, then both are merged to form bigger cluster; if not, a new cluster is created and this object is assigned to that new one. This process continues until all objects (documents/sentences/tweets) are finished. In contrast to RED, NED does not have any initial query or cluster; thus, it must provide some decision rules between new or old events. TF-IDF metric is used in some practices to compare new streams and old ones. In some cases a time attribute is also added to close clusters when specific time is passed; for example, after three days, no further tweets are added to that specific cluster.

"New" and "retrospective" terms belong to *document-pivot* techniques in which algorithms are designed to investigate textual properties of related objects. These techniques aim to provide some metrics to compute similarity of objects based on their textual and linguistic properties.

Being in contradiction to document-pivot methods, *feature-pivot* methods aim to find rapidly growing property in detection stream. This so called *bursty activity* with rising frequency describes a new event fortuity. For example, maybe a huge rise in hashtag usage frequency in Twitter is due to a new event which is happening or has been occurred recently.

Some Twitter event detection and tracking methods use predefined information about users or administrators interests. These methods are known as specified event detectors. Some other techniques do not need any information about events to be tracked and detected and find the real-world occurrences, topics, and events by their properties in frequency raise pick or in terms of similarities. These two distinct methodologies are known as *specified event* and *unspecified event* detection and tracking systems.

As described in this section, many categorizations are drawn for event detector systems; these categorizations lack the main methodology part of algorithms. Section 5.1 describes a new categorization and explains existing methods under this categorization. Table 1 shows a list of methodologies that are studied through this manuscript.

4. Preprocessing

Preprocessing of data in data mining related applications is a common practice while it is also inevitable in the Twitter event detection task. This task includes parts such as data normalization, removal of noisy data, and amendment. NLP tasks require grammatically correct text with certain

Table 1: Twitter topic/event detection/tracking related studies.

Reference	Detection method	Detection type		Detection task		Data collection Dataset	Detection task
		Event	Topic	RED	NED		
[12]	Naïve Bayes classifier		✓		✓	Twitter API, handpicked users	Hot news detection
[13]	BScore based BOW clustering	✓			✓	Twitter API (offline)	Disaster and story detection
[14]	BOW distance similarity	✓			✓	Twitter API	FSD (first story detection)
[15]	BNgram and TF-IDF		✓	✓		Offline datasets	Topic detection
[16]	Cross checking via Wikipedia	✓			✓	Twitter API, Wikipedia	Hot news detection
[17]	Formal concept analysis		✓		✓	RepLab 2013 dataset	Topic detection
[18]	FPM (frequent pattern mining)	✓			✓	Twitter API	Event detection
[19]	FPM		✓	✓		Super Tuesday/FA Cup/US elections	Topic detection
[20]	FPM (hierarchical clustering)		✓		✓	Topic dataset from CLEar system	Topic detection
[21]	FPM (TF-IDF & *n*-gram improved)	✓			✓	Twitter API	Event detection
[22]	GPU improved TF-IDF approximation		✓	✓		Offline dataset	Topic detection
[23]	BOW similarity	✓			✓	Offline dataset	Topic detection
[24]	Word embedding					SemEval dataset	Twitter sentiment classification
[25]	Spatiotemporal detection	✓		✓		Offline dataset	Targeted-domain event detection
[26]	Clustering of temporal & spatial features	✓		✓		Twitter API	Event detection
[27]	Geographical regularity estimation	✓			✓	Twitter API	Geosocial event detection
[28]	BOW clustering	✓			✓	Twitter API	Event detection & analysis
[29]	Probabilistic modeling	✓			✓	Twitter API	Early disaster detection
[30]	FPM	✓		✓		Offline dataset	Event detection
[31]	Heartbeat graph	✓		✓		Super Tuesday/FA Cup/US elections	Topic/event detection
[32]	Enhanced heartbeat graph	✓		✓		Super Tuesday/FA Cup/US elections	Topic/event detection
[33]	Sentence BERT/streaming graph mining		✓	✓	✓	Super Tuesday/FA Cup/US elections	Topic/event detection
[34]	Universal sentence encoder		✓	✓	✓	COVID-19 dataset	COVID-19 topics
[35]	TF-IDF, CCA, and BTM		✓	✓		Twitter API	Trend ranking
[36]	LDA, USE, and SBERT	✓		✓		COVID-19 dataset	COVID-19 topics
[37]	Autoencoder and fuzzy c-means		✓	✓		Berita	Trend ranking

properties. Preprocessing is one of the main parts of social big data analysis subtasks. Short tweets communicated through Twitter service as described before need to be processed to be ready for further event detection computations. Removal of stop words and punctuation marks is a crucial step in preprocessing of natural language processing related data mining tasks [38]. Identification of URLs and emojis is also needed. Regular expressions can be used to detect URLs in short messages.

In some cases, stemming is also applied for unification of processed words while non-target-language words are also vanished in this process. Elimination of non-target-language words helps improve extracted topic to be in a target language. Tokenization is also another part of preprocessing that gives unique tokens to each word in a tweet. This part of preprocessing is more crucial in TF-IDF (Term Frequency-Inverse Document Frequency) related models.

Some methodologies like *EvenTweet* [26] use WordNet [39] check as part of their preprocessing. This WordNet dictionary lookup improves correctness of preprocessing output; thus, no non-English and incorrect words will be used for event detection task. Slang word translation is also used to translate user generated words into their formal meaning. *NoSlang* website is also a common tool for this task [40].

Common information retrieval processes from Twitter or any other online web-based data sources require special preprocessing techniques. One of these techniques is removal of unwanted and trashy character sets such as HTML tags. Sometimes these trashy looking character sets seem to be useful (in case of encoding and critical information related to data). White space and punctuation marks that are also called white spaces need to be sorted out. An example of these occurrences is *Ph.D.* that has ambiguity of end of sentence; another example is *$5.79*.

The main concepts of a clean and clear text are *Word Token* and *Word Type*. The first one refers to occurrences of words that are numbered while the latter one implies unique words that are entries of a table called *vocabulary* list. Tokenizing a text is a natural language processing task aimed at tokenizing words and giving them unique numbers in sentence which later will be used by tasks such as stemming or part of speech tagging.

As discussed so far, preprocessing is an essential and inevitable part of any natural language processing algorithm, and in case of Twitter TDT task it is also demanded.

5. Event Detection and Tracking Task in Twitter

Event detection and tracking task in Twitter is a well investigated research issue. This section provides details of approaches that are applied to this problem.

5.1. Event Detection in Twitter: Methodological Categorization. Event detection and tracking in most of cases is composed of known data mining methods that have been used before in different areas. Such algorithms and methods are combined with NLP techniques to obtain better results over testing process of algorithms. In this subsection we try to categorize existing algorithms for this task with respect to their utilized data mining and NLP methods.

5.1.1. Bag of Words Methods. Inclusive methods of this category mainly use TF – IDF metric to extract final topic related to tweets, and any other features of a sentence like its part of speech tags are disregarded. Term Frequency-Inverse Document Frequency, abbreviated as TF – IDF, is a common metric among most of topic detection or extraction methods and is described as (1) and (2). Respectively, t and d in these equations refer to term and document, which in case of the latter can be assumed as a single document containing more than a tweet, maybe couple of tweets or just a single tweet which can also be referred to as a message. Furthermore, count $(t \text{ in } d)$ represents counting occurrences of term t in document/message d while count $(d \text{ has } t)$ denotes counting documents/messages that have at least one occurrence of t.

A similarity metric is used with utilization of TF – IDF to compare two separate tweets in [41]. This similarity metric described in (3) is used as a score function to group new messages; a message that does not belong to any group is considered to be a new group. New groups are populated in order of classification of new messages with respect to score function. To avoid unrelated messages to first one in a group, all messages are compared to first message and top k messages.

$$\operatorname{tf}(t,d) = \frac{\operatorname{count}(t \text{ in } d)}{\operatorname{size}(d)}, \quad (1)$$

$$\operatorname{idf}(t) = 1 + \log\frac{N}{\operatorname{count}(d \text{ has } t)}, \quad (2)$$

$$\operatorname{similarity}(d_1, d_2) = \sum_{t \in d_1} \operatorname{tf}(t, d_2) \times \operatorname{idf}(t) \times \operatorname{boost}(t). \quad (3)$$

Another method described in [12] represents a new architecture for news related TDT task from Twitter. In this architecture, a cosine similarity measure is utilized along with TF-IDF representation of tweets to accomplish this task. This similarity measure is computed between tweet t and cluster c. Equation (4) shows related mathematical expression. Feature vectors of $\overrightarrow{\mathrm{FV}_t}$ and $\overrightarrow{\mathrm{FV}_c}$ are obtained from TF – IDF model of messages. A Gaussian attenuator is then applied to this similarity measure to place impact of temporal dimension in clustering. This weight makes sure that no old clusters and messages get twisted. This architecture makes use of hand selected users which are most likely to post news and also a sampling and tracking system.

The *BNgram* model that is introduced in [15] along with sentiment classification and part of speech tagging forms a trending topic detection system. *BNgram* model in this research is similar to [41] with small differences that imply boost factor. If this factor is set to 1.5, then n-gram model holds named entity; otherwise, it is a small number, and the respective model does not hold a named entity. Based on *n*-gram TF-IDF, all tweets are scored and, based on these scores, are then clustered into respective clusters. This scoring and clustering process is conducted in time windows, and in each time step, tweets related to a time window are compared to others that have been posted earlier. The proposed method has been trained on some handpicked datasets collected from Twitter API which were related to sports (the Cricket World Cup 2015), medicine (Swine Flu 2015), and bills (Land Acquisition Bill). Compared to frequent pattern mining methods, this method seems to be a simpler algorithm in terms of software implementation with good results in terms of output topics on some cases that shamefully are not expressed as F-measure, precision, recall, or any related metrics. The only social big data tool that this method uses is Lucene for keyword indexing.

“Bieber no more!” is title of another article in these criteria which uses simple nearest neighbor among tweet hashtags to find dissimilarity of previously seen events and new ones [16]. This first story detection system utilizes Wikipedia as a source of information. Wikipedia is a multilingual, web-based, free-content encyclopedia project supported by the Wikimedia Foundation and based on a model of openly editable content. Wikipedia page view helps to find out if an event occurred recently or it is just a false positive detected by this system. Simple use of nearest neighbor among hashtags of multiple tweets and utilization of Wikipedia are expanded to a multistream first story detection system. This system works in the same manner of single-stream first story detection with the only difference being in vector space modeling. This vector space modeling between tweets and Wikipedia pages checks the following: if any new event occurred, it is reflected as pick user page views in Wikipedia; if it was a false positive, no pick view on Wikipedia-related page happens.

Another first story detection system is proposed in [14]. This system makes use of an improved version of locality sensitive hashing (LSH) within a $(1+\varepsilon) \times r$ distance of query point for Twitter first story detection. Time and space bounding narrow nearest neighbor finding problem. This problem arises when huge amount of user tweets are posted per day, and the goal is to find out if they point to a new story/event or a previously seen one; storing all of these data and finding nearest neighbor between them are almost impossible. Time bounding refers to using a time window instead of computing all data from all times while space bounding points to solving this problem among limited number of tweets. Similarity of a tweet compared to previous ones shows if it is new or not, and this task guides proposed system to open a new story or keep it the way it was.

Another way of extracting answers for 4-W question, *Who*, *What*, *When*, *and Where*, is proposed in [42] which uses a new data representation method called *named entity vector*. This data representation vector along with *term vector* is integrated as a *mixed vector* to obtain results.

$$\text{cosine\_similarity}(t, c) = \frac{\overrightarrow{\mathrm{FV}_t} \cdot \overrightarrow{\mathrm{FV}_c}}{\left\|\overrightarrow{\mathrm{FV}_t}\right\| \times \left\|\overrightarrow{\mathrm{FV}_c}\right\|}. \quad (4)$$

Term Frequency-Inverse Document Frequency (TF-IDF), Combined Component Approach (CCA), and Biterm Topic Model (BTM) are the main approaches addressed in [35]. Ranking trends is aimed to be solved by authors by using these models and features.

5.1.2. Probabilistic Models and Classifiers. Probabilistic topics models and classifiers that are described in this section are used to model and classify Twitter datasets or streamlines. One of these approaches that is presented in [23] uses a Naïve Bayesian classifier called NB-Text to satisfy this requirement. This probabilistic method is trained over 2,600,000 Twitter messages annotated by humans posted on 2010. This dataset is labeled for training and testing phases. Firstly, a classifier called RW-Tweet is trained to distinguish between real-world and non-real-world events. Weka toolkit [43] along with extracted cluster level features is used to train classification model. This Naïve Bayesian classifier treats all messages in a cluster as a single document and uses TF-IDF metric as features. Cluster level event features such as temporal, social, Twitter central, and topical features are utilized for this classifier.

TwitterStand is the name of another system proposed in [12] that clusters events by a Naïve Bayesian classifier. This can deal with noise and fragmentation. Noise, according to the authors, is clusters that are not relevant to real-world events; thus, reliable news sources as seeds are used instead of regular users, which weakens this system. This assumption is true when news sources post news in real-time, but the nature of social media has proven that users are the real people who happen to be a part of event or disaster. On the other hand, fragmentation refers to duplicate clusters that mean the very same event. Periodic checking of duplicate clusters overcomes this problem on the system. Event geolocating of this system makes it stronger and more useful.

5.1.3. Formal Concept Analysis. Formal concept analysis has been used by [17] in an unsupervised fashion. RepLab 2013 dataset [44] is used to evaluate this system. Formal concept as it is known from literature is an approach for finding relations between data that is almost hidden in its nature. This relation can be defined between objects and their attributes.

Extent: if we see A as a set of objects (itemset), then it is called an extent

Intent: if B is a set of all attributes of set A, then it is called intent

Formal concept analysis in this way is formalization of extension and intention to find the most related items that possess important attributes in share.

In [17], tweets are seen as objects and their terms are attributes, which makes this methodology very similar to the ones described in Section 5.1.4 as FPM methods. The proposed method tries to find *concept lattices* in unstructured data of tweets, which shows good reliability and sensitivity. A set of tweets in proposed setup of this work are assumed to be objects while terms (words) are attributes. A relation indicates that a term has been used in a tweet. Formal concepts extracted from concept lattice show topics. Some of these concepts are discarded to have better topic. Small concept lattice and terms are computable with this methodology while bigger size of corpus and tweets and vast number of terms lead to a huge lattice. In such a case, a term selection strategy is required to narrow down this problem. Most shared attribute selection strategies drop least shared attributes (terms). This balanced version of algorithm utilizes term frequency of each attribute. This term frequency (tf) shows a threshold of selecting which term should be used in concept lattice. In each iteration, terms with highest tf are selected, and objects (tweets) with less than two terms in their attributes are discarded. Last iteration of this fine-tuned strategy outputs the attributes with highest tf and objects that possess them. Last step of this framework is to actually make topics out of these lattices. However, the previous step has reduced the potential concept lattices to be candidates of final topic. Stability concept that has been previously proposed in [45] indicates how much concept intent depends on objects available in extent. This reduction with keeping stability helps to form topic.

5.1.4. Frequent Pattern Mining Methods. Frequent pattern mining methods have been applied to TDT task in Twitter. Frequent pattern mining (FPM) as indicated by its name is concept of finding frequent itemsets in a database or any related data storage. A simple example of these frequently repeated patterns is described as a set of coffee and donuts which are in most of cases bought together [46].

In [19], a FPM algorithm is introduced for Twitter offline dataset and compared to other relative studies. FP-growth algorithm with small modifications and utilization of similarity metric is applied to form a set of related tweets that form a topic. Cooccurrence patterns between terms that are larger than two constitute main contribution of this work. Three phases of topic extraction in this method are term selection, cooccurrence-vector formation, and post-processing. First stage indicates that likelihood of terms occurrence in a corpus is major concern. A probability such as $P(\text{term}|\text{corpus})$ is obtained in this phase, and between a new corpus and this reference corpus, this likelihood is compared with ratio of $(P(\text{term}|\text{corpus}_{\text{new}}))/(P(\text{term}|\text{corpus}_{\text{ref}}))$. This ratio is a metric to show how a term frequency is changing. Higher ratio means higher frequency of appearance, and thus this term can appear in the final topic. Next phase constructs S and D matrices that are later used for frequent pattern mining. Matrix D_s shows how many terms of S appear in several documents while D_t shows how many times a term appears in several documents. Cosine similarity between these two matrices indicates how a term is suitable for adding to final topic. A sigmoid function is used to limit this similarity and act like a threshold. Final phase of this algorithm is a cleaning stage to remove duplicated topics.

Moreover, a similar method that uses FPM to detect social events from Twitter is introduced in [21]. At first step, the K most relevant terms of current set of tweets such C_{curr} are selected by means of highest appearance likelihood. After this step, the soft version of FPM with utilization of sigmoid function as a threshold computes similarity. Social aspects such as event, spam, and past event are introduced to evaluate performance of system. This system performs on live Twitter streamline.

The idea of burst pattern mining that is introduced in [20] is used to construct burst topic user graph with other various features. These features are *tweet number*, *retweet ratio*, *reply ratio*, *user number*, *overlap user ratio*, *big user ratio*, *burst number*, *burst interval*, and *burst time interval*. Macro and micro burst patterns are defined as bellow as main contributions of this work.

Macro burst pattern is finding all clusters in BT in which BT is a burst topic set, and this task is accomplished with the use of a distance measure among features.

Micro burst pattern is finding all subgraphs in user graph G such that $\sup_G(GS) >$ treshold.

This algorithm starts with finding set S that contains all frequent edges, and with use of DFS (Deep First Search algorithm), the subgraph extention algorithm eliminates nodes that do not satisfy the support threshold (τ). The subgraph extention algorithm is executed recursively to extend frequent subgraphs.

Association rule mining (ARL) is another approach of frequent pattern mining in relational databases that has been used in [18] to detect events in Twitter. ARL has two parts: antecedent and consequent. An antecedent is an item that is found in data while a consequent is an item found in combination with the antecedent [47]. These can be named as if/then (antecedent/consequent) patterns with help of criteria support to identify the strongest and most important relations between items in data. In [18], two main equations are used to match rules with regard to their similarity; they are adopted from [48]. Emerging rules as a contribution of this work are proposed to identify breaking news. US Elections dataset has been used to evaluate the proposed methodology that shows good results in terms of F-measure, recall, and precision.

Tracking dynamics of words in terms of graph, or converting sentences into graph representation and trying to understand the spikes inside, is a very useful method. The graph heartbeat model, introduced by [31], and its enhanced version [32] are all based on this fact. They used graph analytics to detect the emerging events from Twitter data stream by using graph based formulation and spike detection. This spike detection that is called heartbeat model is a mathematical formulation of matrix analysis during detection of events from Twitter social media.

5.1.5. Signal Transformation-Based Approaches. Signal transformation based approaches, such as *Fourier* and *wavelet* transforms, apply spectral analysis techniques to categorize features for different event properties. DFT (discrete Fourier transform) methodology that has been applied in [49] converts burst in time domain to spike in frequency domain. This spike only shows a bursty event, not its period. Thus, a mixture of Gaussian models for identifying time period of these feature bursts have been applied. Fourier transform is given in (5) which is invertible, and its inverse transform that leads to the $y_f(t)$ function is given in (6).

$$X_k = \sum_{t=1}^{T} y_f(t) e^{-((-2\pi i)/T)(k-1)t}, \quad k = 1, 2, \ldots, T, \tag{5}$$

$$y_f(t) = \frac{1}{T} \sum_{k=1}^{T} X_k e^{(2\pi i/T)(k-1)t}, \quad t = 1, 2, \ldots, T. \tag{6}$$

With these prerequisites known, the dominant period spectrum can be explained further; this period is assumed to be a period in which the specified frequency reaches its maximum activeness or, in other words, it is bursty. These specifications tempted the authors of [49] to categorize all features into four main types, *HH, HL, LH, and LL* (the first letter shows Dominant Power Spectrum, and the second letter indicates dominant period in which H means high and L means low). Detecting periodic feature bursts is accomplished by aid of a Gaussian mixture.

Reference [30] presented a new online event detector in news streams with utilization of statistical significant tests of n-gram word frequency within a time frame. Three definitions given in the original manuscript are *textual data stream*, *alphabet*, and *time frame* that are, respectively, described as a sequence of text samples S_t that is sorted by t (time), English words (such as "president" and "coffee"), and a time range starting from t_0 and ending at T in form of $[t_0, t_0 + T]$. In this terminology, an event is described to be a change in the source of text stream which is a surprising rise in n-gram frequency. Computed p value for n-gram hypotheses gives a clear insight about the correctness of the null hypothesis that is stated to be "*two individual textual datasets of two time frames are generated from one source.*" Due to vast variety of n-grams, a suffix tree is also proposed to store the n-gram. Computed frequency is stored in this new data structure, and another algorithm runs over the tree to calculate and store p values along with it.

Clustering of discrete wavelet signals of words generated from Twitter is also another approach that is used in [50]. Unlike Fourier transform, wavelet transformations are localized in both time and frequency domain and hence able to identify the time and the duration of a bursty event within the signal. Wavelet signal transformation transforms signal from time domain to time and scale domain. A wavelet family is defined in

$$\psi_{a,b}(t) = |a|^{-1/2}\psi\left(\frac{t-b}{a}\right), \quad (a,b \in \mathbb{R}),\ a \neq 0. \tag{7}$$

Wavelet energy, entropy, and H-measure are also other discrete wavelet transformation parts that give useful information about the signal. H-measure is normalized Shannon wavelet entropy that shows distribution of signal over different scales. The proposed *EDCoW* algorithm (Event Detection with Clustering of Wavelet-based signals) has three main components of signal construction, cross correlation computation, and modularity-based graph partitioning.

First step computes DF-IDF (DF is not TF and it means document frequency rather than term frequency) shown in the following equation:

$$\mathrm{df} - \mathrm{idf}(\mathrm{term}) = \frac{N_{\mathrm{tweets}}(\mathrm{term})}{N(\mathrm{term})} \times \log \frac{\sum_{i=1}^{T_c} N(i)}{\sum_{i=1}^{T_c} N_{\mathrm{Tweets}}(i)}. \tag{8}$$

A raise of DF-IDF metric is also reflected as a raise in wavelet entropy of this metric. Cross correlation of two different signals is used to group words/terms that happen to have raise in their wavelet entropy together, meaning that these terms have been used together in a topic that previously seen in a raise or happened to be an event candidate. This clustering methodology is suitable for signal transformed detection. A modular sparse matrix is formed at the last phase of this work to detect events by clustering the weighted matrix. This matrix is called M and is in form of $G(V,E,W)$ in which V is vertices, E is edges, and W is weights of the graph G.

A similar method is [51] which uses LDA and hashtag occurrences. This method, unlike [50], uses hashtags to build wavelet signals. LDA is used to form the final topic model. Another difference between this work and [50] is summarization of extracted events that is done with the aid of LDA topic inference and seems to show promising results but cuts off the tweet data and reduces it to hashtags. This reduction harms the algorithm but improves its speed compared to the latter one.

5.1.6. Geoevent Detection Methods. Methods that are described earlier try to only answer the question "What is happening?" However, there is another question yet to be answered: "Where it happened?" Geolocation of an event expresses more insights of a detected event. In [25], a spatiotemporal event detection scheme is proposed; it detects events along with their occurrence time and also geolocation. Some definitions need to be known before further description of algorithm; these definitions are *spatiotemporal event* and *article*.

Spatiotemporal event is a real-world incident that happened at location l and time t which is denoted by $\mathrm{event}_{l,t}$. Domain is known to be set of events that fit into a categorization such as music and civil.

Article set of targeted domains can be open or closed. A closed article such as A_p denotes an article related to topic p, and a_x can be a news report from that article.

This manuscript suggests two types for tweet categorization in order to classify tweets as related/unrelated to event. A *positive tweet* is a related tweet to event, and in contrast a negative tweet is simply an unrelated tweet to the event. With all this setup, we can dive into the concept of *label*. A tweet label is known to be a triple of $z = (x, Y^{(x)}, \widehat{Y}^{(x)})$, where x indicates event, $Y^{(x)}$ indicates related tweets, and $\widehat{Y}^{(x)}$ expresses unrelated tweets. Label generation is task of classifying labels of specific topic that are also related/unrelated to the event. After this step is completed, the next step of proposed work is spatiotemporal event detection. This last step inputs a label set on a specific topic that is given from previous step and the real-time Twitter stream and outputs the online event sets of targeted domain that are happening or happened in location l at time t.

First step of this work consists of feature extraction and relevancy ranking. The relevancy ranking step ranks tweets based on how they are relevant to event in terms of textual and spatial similarity. These ranked features are then used by a tweet classifier that is a SVM-based (Support Vector Machine) classifier. Event location estimation is the latest step of this scheme to estimate actual location of classified tweets.

TEDAS is another spatiotemporal event detection system originally proposed in [28]. This system has three main phases: detecting new events, ranking events according to their importance, and generating spatial and temporal patterns of detected and highest ranked events. Java and PHP along with MySQL are utilized to make this system that also makes use of Lucene, Twitter API, and Google Maps to output final user friendly output. Crime and disaster related tweets are subject to this system. A query based use of Twitter API has been applied to obtain tweets. A set of rules for query are needed, so some simple rules are used to obtain tweets, and later these rules are populated with the help of obtained tweets. Twitter and crime or disaster based features help the next phase of this system to classify the obtained tweets; this classifier has accuracy of 80% as authors indicate. The last phase of this scheme uses content, user, and usage related features to rank the detected events while previous phase is focused on guessing the location of user. The first assumption is that the location of user is in his GPS-tagged tweets if there are any; if not, his/her friends are more likely to be close to him. The last assumption says his/her location is mentioned in his tweets for at least once. One of the main problems of this location guessing is that in the case of second and third assumptions, the extracted information can be false.

The idea of social sensors that has been used in [29] is proposed to find the location of real-world disasters in Twitter. The definition of event according to the authors is an arbitrary classification of a space/time region. As the earlier method, this scheme also makes use of SVM as classifier with three features of types A, B, and C that, respectively, are known as statistical, keyword, and word context features. Each tweet is known to be a sensory value, and users are the sensors of this scheme. They tweet about the event, meaning that they are sensors and sensed values are posted as tweets. This report is helpful to detect the real-world disasters such as earthquakes. Real problem of this assumption is that there is a possibility of error when a user posts unrelated tweets that seem to be relevant; an example of these according to authors can be this tweet: “My boss is shaking hands with someone!” Shaking as a primary keyword is used in this tweet but it does not mean that the Earth is shaking. Other features of previous part make error possibility lower, but still there is a chance. Two spatial and temporal models are proposed to clarify the assumptions. These models rely on tweet time stamp and GPS stamp. The evaluation and experimental results show that the system shows over 60 percent accuracy on two related queries. This valuable system is used as an earthquake warning system in Japan that in time can save lives of several people.

5.1.7. Deep Learning-Based Methods. Transfer learning in deep learning and specially NLP by using new methods and approaches such as Transformers enabled researchers to use pretrained models for various problems. Topic detection and tracking from Twitter is also one of these problems that researchers tried to solve by transfer learning based models such as BERT. TopicBERT is one these methods that utilizes BERT for semantic similarity combined with streaming graph mining [33]. The proposed architecture is composed of a deep named entity recognition model [52], a graph database to store the nodes, and a semantic similarity extraction tool (SBERT). The whole system works in a combined manner in which the different parts constantly try to update the underlying graph database, and an extraction system using probability of clusters and probability of words gets the topics at each moment. This system beats state-of-the-art methods on three different datasets and is one of first methods that used Transformers for topic detection and tracking from Twitter.

Combination of semantic vector representation of tweets with clustering algorithms is another methodology that is investigated in [34]. The authors show that utilization of a good semantic feature extractor in form of a dense vector can be quiet useful when dealing with problems such as topic detection. They have used COVID-19 dataset from Twitter and detected topics relatively. Another similar method for COVID-19 is proposed in [36]. The authors propose to use Sentence BERT and Universal Sentence Encoder (USE) for sentiment analysis in combination with LDA based topic detection.

Autoencoder based fuzzy c-means algorithm is presented by [37]. Autoencoder is used for representation of tweets while fuzzy c-means is the clustering part of method. The authors report their results on Berita dataset which is an Indonesian news dataset from Twitter.

Utilization of these methods, which are all based on deep learning, is a new field in NLP, specially transfer learning based ones that use Transformers to have a semantic understanding of text. This semantic understanding is a missing part of other methods. The semantic clustering used by various methods can categorize texts with different words into a single cluster if they have close meaning. Language models and pretrained transformer based architectures that can capture semantic similarity such as SBERT and USE are successful examples of these approaches. These approaches are well known for their ability to understand complex sentences. In case of USE, it can even match sentences from different languages to each other if they carry the same semantic meaning. Compared to non-deep learning based methods, these approaches provide a semantic way to TDT task in Twitter.

5.1.8. Performance Improvements. Recently modeling data as image and processing it on graphic cards constitute a useful view to fasten data processing and obtain real-time or at least near real-time results. As it has been described before, TF-IDF has been used widely used for TEDT task. Methodology of fastening data processing presented in [22] uses an approximation way to figure out the TF-IDF metric. Similar to FPM methods (Section 5.1.4), it uses a sort-based algorithm to find frequent items (tweets). The described algorithm is inspired by [53].

The first step of this algorithm is to find the most frequent itemsets. If we assume that set of B contains all of ordered pairs, the next step is to reduce these items by their id or just simply add the pairs that have the same id. The last step would be to divide them to total count of itemsets, and the result would be TF. The whole process of this algorithm can be run in parallel on a dedicated GPU which gives it more computing power than regular CPUs and is more suitable for real-time computation of TDT task, because other algorithms are weak on this aspect and most of them are applicable to offline datasets.

5.1.9. Deep Learning Short Sentence Sentiment Classification: A Post-TEDT Phase. The main difference of algorithms and machine learning methods described in this section is that they do not detect topics or track events on Twitter. Instead, they can be recommended after event or topic detection phase in which the overall sentiment of users is averaged on the detected topic. This output can give great analytical information. Algorithms, machine learning roadways, and neural networks categorized in this subsection are post-topic/event detection step with regard to deep learning.

Recently, with emerging growth of deep learning methods in NLP tasks, short sentence classification and sentiment analysis of these sentences have seen a major change of methods and applications. Deep learning, as suggested by its name, allows computational models to have a lot of abstraction layers for data representation [54]. Raise of unified architectures of multilayer neural networks for NLP tasks seems to be a promising methodology to solve many unsolved problems in this scope [55] while word embeddings such GloVe [56] and Word2Vec [57] suggest new vector representation of words that also possess sentimental property of dedicated words and can be applied in terms of matrix calculus.

Sentiment analysis of short sentences has been focused on by many researcher from many aspects such as short sentences (CharSCNN) [58]. On the other hand, distinct characteristics of corpora obtained from Twitter led researchers to find new algorithms of sentiment analysis and sentence classification tasks in Twitter which are foundation of topic and event detection in Twitter using these new research outcomes.

Like other word embedding algorithms, CharSCNN in its first layer transforms the input words into encoded vectors representing distinct words. Any word such as W that has been encoded into a vector in previous layers is separated in terms of its characters, and each character is encoded into another vector such as r_m^{chr}. Matrix vector multiplication of set $\{r_1^{\text{chr}}, r_2^{\text{chr}}, \ldots, r_n^{\text{chr}}\}$ gives r^{chr} for each character that would be character embedding in this layer. Sentence level representation and scoring are applied as described in character and word level. CharSCNN has been applied to two distinct short sentence datasets of Movie Reviews and Twitter posts with word embedding size of 30.

Sentiment-specific word embedding for Twitter sentiment classification that is proposed in [24] uses *C&W* method of [59]. Three different neural networks (SSWE_h, SSWE_r, and unified model of SSWE_u) are proposed in this manuscript for different strategies to overcome task of Twitter sentiment classification.

5.2. Specified versus Unspecified. Based on available information about an event that is to be detected, an event detection method can be categorized as specified or unspecified. Unspecified methods mainly rely on detecting temporal signs of Twitter such as bursts or trends. These methods have no prior information about an event, and thus they need to classify relative events based on bursty properties and cluster them. Specified event detection systems, unlike previous ones, need some information of an event that can be its occurrence time, type, description, and venue. These features can be exploited by adapting traditional information retrieval and extraction techniques (such as filtering, query generation and expansion, clustering, and information aggregation) to the unique characteristics of tweets. The next subsections categorize existing methods based on this terminology.

5.2.1. Unspecified Event Detection. User driven Twitter short posts sometimes contain very important information about real-world events that are published by users before news media websites and TV/radio channels. These short but important posts are unknown to event detector system and also not predefined by any supervisor. A raise in Twitter temporal and signal patterns can reveal this fact. For example, a sudden and unexpected raise in use of a keyword or hashtag may show a sudden attraction to that topic, and somehow that might reveal a real-world event. An ambiguity occurs due to this setup while some frequent hashtags and keywords about daily life tweets are detected as unseen and new event. An efficient unspecified event detection algorithm must deal with this kind of ambiguity.

In [60], an event detection system called TwitterMonitor is proposed. TwitterMonitor identifies emerging topics in real-time in Twitter and provides meaningful analytical information that can be further used to extract a topic to detected event. A StreamListener listens to Twitter API data stream and detects bursty keywords; these keywords are then grouped and along with an index are passed into Trend Analysis module. All of described steps form the backend of system while a user interface sums up all of information and presents it to user. Other implementations such as AllTop, Radian6, Scout Lab, Sysomos, Thoora, and TwitScoop have a user interface to represent information gathered from different social media, newswire, and other data stream lines to the front end user.

TwitterStand is another electronic medium that, with use of Naïve Bayes classifier, separates news from irrelevant user generated tweets [12]. Cosine similarity metric along with TF-IDF weighing classifies the cleansed events. A breaking news detection system also fits this scope that has been previously described [41]. This method collects, groups, ranks, and tracks breaking news from Twitter by sampling tweets and indexing them using Apache Lucene.

First story detection (FSD) system proposed in [14] uses a thread based ranking algorithm to assign a novelty score to tweets and then clusters tweets based on cosine similarity between them. Each tweet is assigned to a thread if it is close to tweets in that thread; otherwise, a new thread is made for this new category. The bigger similarity threshold results in thin categories that are mostly the same while lower threshold results in fat threads.

5.2.2. Specified Event Detection. Specified event detection terminology deletes the question "What is happening?" It simply tends to find "where" or "when" it is happening. The first part of query is known to system, and the latter parts are yet to be answered.

Researchers of Yahoo! Labs in [61] tried to find controversial events that users tend to disbelieve or have opposing opinions about. Controversial event detection is process of detecting events and ranking them according to their controversy. The authors proposed three models for this task: direct model, two-step pipeline model, and two-step blended model. Direct model scores event based on a machine learning regression based algorithm, two-step pipeline model detects events from the snapshot and then scores them based on the controversy, and the soft model of the described one is the two-step blended model. Twitter based news buzz and news and web controversy features are the main feature classes used by this system. This system is user negative opinion mining rather than an event detection system while it still detects events based on entity query.

The very same authors of [61] described another system in [62] that also extracts descriptors from Twitter about the events. Gradient boosted decision trees in a supervised machine learning fashion are employed to form two main models that authors described: EventBasic and EventAboutness.

Many other methods that are categorized as in this subsection are described earlier and are put together in a cumulative manner in Table 1.

5.3. Unsupervised versus Supervised. Machine learning algorithms are trained in both supervised and unsupervised fashions. This means that a training task can be accomplished using labeled data and the machine learning algorithm is assigned to learn the labels from tagged data, while in the unsupervised methodology, it is accomplished by learning by categorization of unknown data labels that are later to be scored. The unsupervised machine learning algorithms have harder job to do in terms of learning with unknown labels. This subsection describes the unsupervised and supervised algorithms for Twitter TEDT task; other algorithms that are described in previous sections are discarded.

5.3.1. Unsupervised Algorithms. Twitter event detection algorithms that use unsupervised machine learning concepts mostly rely on clustering algorithms. As was described earlier, NED is a term used to identify new event detection systems that, contrary to RED (retrospective event detection), detect and identify new events, while the latter one detects and identifies specified events. Unsupervised methods are highly recommended for tasks that require clustering of unknown categories that exactly fit the NED domain. Furthermore, there is no prior information about the number of classes to be categorized because of dynamic nature of user activities in social networks.

5.3.2. Supervised Algorithms. Supervision of a clustering algorithm that needs labeled data to classify the user generated real-life events has a close relation to RED category. As described earlier, the RED algorithms tend to classify the known events while supervision needs labeled data in its training phase. This terminology has many shortcomings in real-world applications such as event detection system. A system that is aimed to find and track real-world incidents cannot be trained in supervised fashion; this is because of unknown events that yet to come and absence of information about their quantity and entity.

6. Data and Evaluation Issues

Twitter by its nature possesses unstructured and unlabeled data stream that can be obtained from online or offline sources. Online Twitter data source is the Twitter API, and offline data is the offline Twitter data obtained from different snapshots. These snapshots possess better properties to evaluate differences between algorithms or systems that aim to find events or topics on Twitter. Evaluation of an online Twitter event extraction system is doable if the input data is the same input data snapshot that is recorded.

Another drawback of event detection and tracking algorithms that has indirect relation to the previous issue, is the event detection time. Suppose that two algorithms or systems such as A_1 and A_2 both have the same precision and recall on finding events and tracking them on Twitter data snapshot but have different detection times. Detection time is defined as the time it takes for a typical algorithm to detect and identify events and track them. If these times (that is related to time complexity) are the same, we can assume that both algorithms are the same, but in case of different times, the near real-time algorithm should be used and preferred. This metric is not reported in any of the works that have been studied in this manuscript, but it seems an essential step to define a real-time event detection and tracking system. In the case of offline systems, this metric is not important.

Both of the evaluation issues described earlier heavily affect the process of evaluation. The Defense Advanced Research Projects Agency (DARPA) published the results of a competition named "The DARPA Twitter BOT Challenge" [63]. The contestants of this competition were the big companies of information technology industry (SentiMetrix, IBM, USC, DESPIC, B. Fusion, G. Tech). A mathematical scoring system was used to score the bots created by contestants. Equation (9) defines this scoring system. This competition aimed to create bots that can identify fake users (bots) that are posting on Twitter and creating influence. However, the relevance of this research is important, and it is related to event detection and tracking system because the scoring system used in this competition is a usual artificial intelligence related measuring system which also points to speed.

$$\text{Final Score}(t) = \text{Hits}(t) - 0.25 \times \text{Misses}(t) + \text{Speed}. \quad (9)$$

A related scoring system to event detection systems according to (9) can be extracted. The very same manner of speed in evaluation of event detection system is also used in [64] to measure quality of systems.

Duplicity of detected events or topics is also another drawback. Misdetection of events and identification of a nonevent phenomenon also constitute a huge problem. The reason this issue possesses bigger threads is that a real-time disaster informing system can be fooled and misdetect a disaster or even not detect it.

With all of these in mind, an evaluation/scoring system for TEDT requires quantities of HITS, MISSES, recall, precision, and speed to be calculated on a specific data snapshot of Twitter. Otherwise, the systems cannot be compared to each other. A typical scoring system can be known as 10 with α, β as weights. Other scores of Score_2 and Score_3 are the precision and recall of algorithm on the dataset.

$$\text{Score}_1(t) = \alpha \times \text{Hits}(t) - \beta \times \text{Misses}(t) + \text{Speed}. \quad (10)$$

7. Conclusion

Twitter as one of the biggest social networks and microblogging services enables users to post and share their thoughts, daily life posts, and news about real-world events. Many of these users' posts are related events are real-world incidents and some are rumor, meaningless, and plot information. Unfolding these real-world events and extracting them from Twitter need real-time systems with high accuracy and precision. Evaluation of systems faces many issues such as data and evaluation metric problems. In this article, we studied some TEDT systems that aim to find, detect, extract, and track real-world incidents from Twitter and also described the problems related to evaluating such systems. Many categorizations were proposed to classify these algorithms and methods that are also presented in this article; in addition, another categorization based on the methodology of the relying algorithms is proposed in this article. Finally, this article discussed a postdetection methodology proposed as deep learning short sentence classification that can be useful after detection of events.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Allan, *Introduction to Topic Detection and Tracking*, Springer, Berlin, Germany, 2002.

[2] J. Allan, *Topic Detection and Tracking: Event-Based Information Organization*, Vol. 12, Springer Science & Business Media, Berlin, Germany, 2012.

[3] J. Allan, J. G. Carbonell, D. George, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study final report," in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pp. 194–218, Lansdowne, VA, USA, February 1998.

[4] Twitter Usage Statistics, 2017, InternetLiveStats.com.

[5] Twitter Tweets Per Day Statistics, 2013, https://blog.twitter.com/2013/new-tweets-per-second-record-and how.

[6] M. James, M. Chui, B. Brown et al., *Big Data: The Next Frontier for Innovation, Competition, and Productivity*, McKinsey Global Institute, New York, NY, USA, 2011.

[7] G. Bello-Orgaz, J. J. Jung, and D. Camacho, "Social big data: recent achievements and new challenges," *Information Fusion*, vol. 28, pp. 45–59, 2016.

[8] A. Białecki, R. Muir, and I. Grant, "Apache lucene 4," in *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval*, Portland, OR, USA, August 2012.

[9] UIMA Apache, Apache Software Foundation, 2011, https://java.apache.org.

[10] Apache, Apache Storm, 2013.

[11] MongoDB, Mongodb, 2013.

[12] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, "Twitterstand: news in tweets," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 42–51, ACM, Seattle, WA, USA, January 2009.

[13] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in twitter," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference onWeb Intelligence and Intelligent Agent Technology (WI-IAT)*, pp. 120–123, IEEE, Toronto, Canada, August 2010.

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

[15] S. D. Tembhurnikar and N. N. Patil, "Topic detection using bngram method and sentiment analysis on twitter dataset," in *Proceedings of the 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, pp. 1–6, Noida, India, September 2015.

[16] M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, and I. Ounis, "Bieber no more: first story detection using twitter and wikipedia," in *Proceedings of the SIGIR 2012 Workshop on Time-Aware Information Access*, Portland, OR, USA, August 2012.

[17] J. Cigarrán, Á. Castellanos, and A. García-Serrano, "A step forward for topic detection in twitter: an FCA-based approach," *Expert Systems with Applications*, vol. 57, pp. 21–36, 2016.

[18] M. Adedoyin-Olowe, M. M. Gaber, C. M. Dancausa, F. Stahl, and J. B. Gomes, "A rule dynamics approach to event detection in twitter with its application to sports and politics," *Expert Systems with Applications*, vol. 55, pp. 351–360, 2016.

[19] G. Petkos, S. Papadopoulos, L. Aiello, S. Ryan, and Y. Kompatsiaris, "A soft frequent pattern mining approach for textual topic detection," in *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, p. 25, June 2014.

[20] G. Dong, W. Yang, F. Zhu, and W. Wang, "Discovering burst patterns of burst topic in twitter," *Computers & Electrical Engineering*, vol. 58, pp. 551–559, 2017.

[21] S. Gaglio, G. Lo Re, and M. Morana, "Real-time detection of twitter social events from the user's perspective," in *Proceedings of the 2015 IEEE International Conference on Communications (ICC)*, pp. 1207–1212, IEEE, London, UK, June 2015.

[22] U. Erra, S. Senatore, F. Minnella, and G. Caggianese, "Approximate TF–IDF based on topic extraction from massive message stream using the GPU," *Information Sciences*, vol. 292, pp. 143–161, 2015.

[23] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: real-world event identification on twitter," in *Proceedings of the Fifth International Conference on Weblogs and Social Media*, vol. 11, pp. 438–441, Barcelona, Spain, July 2011.

[24] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 1555–1565, Baltimore, MD, USA, June 2014.

[25] H. Ting, F. Chen, L. Zhao, C. T. Lu, and N. Ramakrishnan, "Automatic targeted-domain spatiotemporal event detection in twitter," *GeoInformatica*, vol. 20, no. 4, pp. 765–795, 2016.

[26] H. Abdelhaq, C. Sengstock, and M. Gertz, "EvenTweet: online localized event detection from twitter," *Proceedings of the VLDB Endowment*, vol. 6, no. 12, pp. 1326–1329, 2013.

[27] R. Lee and K. Sumiya, "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*, pp. 1–10, ACM, San Jose, CA, USA, November 2010.

[28] R. Li, K. H. Lei, R. Khadiwala, and K. C. C. Chang, "Tedas: a twitter-based event detection and analysis system," in *Proceedings of the 2012 IEEE 28th International Conference on*

Data Engineering (ICDE), pp. 1273–1276, IEEE, Arlington, VA, USA, April 2012.

[29] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 851–860, ACM, Raleigh, NC, USA, April 2010.

[30] T. Snowsill, F. Nicart, M. Stefani, T. De Bie, and N. Cristianini, "Finding surprising patterns in textual data streams," in *Proceedings of the 2010 2nd International Workshop on Cognitive Information Processing (CIP)*, pp. 405–410, IEEE, Elba, Italy, June 2010.

[31] Z. Saeed, R. A. Abbasi, A. Sadaf, M. I. Razzak, and G. Xu, "Text Stream to temporal network—a dynamic heartbeat graph to detect emerging events on twitter," in *Proceedings of the Advances in Knowledge Discovery and Data Mining in Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 534–545, Springer, Melbourne, Australia, June 2018.

[32] Z. Saeed, R. A. Abbasi, I. Razzak, O. Maqbool, A. Sadaf, and G. Xu, "Enhanced heartbeat graph for emerging event detection on twitter using time series networks," *Expert Systems with Applications*, vol. 136, pp. 115–132, 2019.

[33] M. Asgari-Chenaghlu, M. R. Feizi-Derakhshi, L. Farzinvash, M. A. Balafar, and C. Motamed, "Topicbert: a transformer transfer learning based memory-graph approach for multimodal streaming social media topic detection," 2020, https://arxiv.org/abs/2008.06877.

[34] M. Asgari-Chenaghlu, N. Nikzad-Khasmakhi, and S. Minaee, "Covid-transformer: detecting trending topics on twitter using universal sentence encoder," 2020, https://arxiv.org/abs/2009.03947.

[35] H. U. Khan, S. Nasir, K. Nasim, D. Shabbir, and A. Mahmood, "Twitter trends: a ranking algorithm analysis on real time data," *Expert Systems with Applications*, vol. 164, Article ID 113990, 2021.

[36] K. Garcia and L. Berton, "Topic detection and sentiment analysis in twitter content related to Covid-19 from Brazil and the USA," *Applied Soft Computing*, vol. 101, Article ID 107057, 2021.

[37] H. Murfi, N. Rosaline, and N. Hariadi, "Deep autoencoder-based fuzzy c-means for topic detection," 2021, https://arxiv.org/abs/2102.02636.

[38] J. Leskovec, A. Rajaraman, and J. David Ullman, *Mining of Massive Datasets*, Cambridge University Press, Cambridge, UK, 2014.

[39] G. A. Miller, "WordNet," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[40] NoSlang.com, 2017.

[41] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in twitter," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology—Volume 3, WI-IAT '10*, pp. 120–123, IEEE Computer Society, Washington, DC, USA, August 2010.

[42] G. Kumaran and J. Allan, "Text classification and named entities for new event detection," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 297–304, ACM, Sheffield, UK, July 2004.

[43] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[44] E. Amigó, J. C. De Albornoz, I. Chugur et al., "Overview of replab 2013: evaluating online reputation monitoring systems," in *Proceedings of the Lecture Notes in Computer Science in International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 333–352, Springer, Valencia, Spain, September 2013.

[45] S. O. Kuznetsov, "On stability of a formal concept," *Annals of Mathematics and Artificial Intelligence*, vol. 49, no. 1–4, pp. 101–115, 2007.

[46] C. C. Aggarwal and J. Han, *Frequent Pattern Mining*, Springer, Berlin, Germany, 2014.

[47] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, "Algorithms for association rule mining—a general survey and comparison," *ACM Sigkdd Explorations Newsletter*, vol. 2, no. 1, pp. 58–64, 2000.

[48] D. R. Liu, M. J. Shih, C. J. Liau, and C. H. Lai, "Mining the change of event trends for decision support in environmental scanning," *Expert Systems with Applications*, vol. 36, no. 2, pp. 972–984, 2009.

[49] H. Qi, K. Chang, and E. P. Lim, "Analyzing feature trajectories for event detection," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 207–214, ACM, Amsterdam, The Netherlands, July 2007.

[50] J. Weng and B. S. Lee, "Event detection in twitter," in *Proceedings of the Fifth International Conference on Weblogs and Social Media*, vol. 11, pp. 401–408, Barcelona, Spain, July 2011.

[51] M. Cordeiro, "Twitter event detection: combining wavelet analysis and topic inference summarization," in *Proceedings of the Doctoral Symposium on Informatics Engineering*, Porto, Portugal, January 2012.

[52] M. Asgari-Chenaghlu, M. R. Feizi-Derakhshi, L. Farzinvash, M. A. Balafar, and C. Motamed, "A multimodal deep learning approach for named entity recognition from social media," 2020, https://arxiv.org/abs/2001.06888.

[53] U. Erra and B. Frola, "Frequent items mining acceleration exploiting fast parallel sorting on the GPU," *Procedia Computer Science*, vol. 9, pp. 86–95, 2012, Proceedings of the International Conference on Computational Science, ICCS.

[54] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[55] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–167, ACM, Helsinki, Finland, June 2008.

[56] J. Pennington, R. Socher, and C. D. Manning, "GloVe: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October 2014.

[57] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[58] C. N. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 69–78, Dublin, Ireland, August 2014.

[59] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

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

[61] A. M. Popescu and M. Pennacchiotti, “Detecting controversial events from twitter,” in *Proceedings of the 19th ACM International Conference on Information and knowledge Management*, pp. 1873–1876, ACM, Toronto, Canada, October 2010.

[62] A. M. Popescu, M. Pennacchiotti, and D. Paranjpe, “Extracting events and event descriptions from twitter,” in *Proceedings of the 20th International Conference Companion on World Wide Web*, pp. 105-106, ACM, Hyderabad, India, March 2011.

[63] V. S. Subrahmanian, A. Azaria, S. Durst et al., “The darpa twitter bot challenge,” *Computer*, vol. 49, no. 6, pp. 38–46, 2016.

[64] A. Weiler, M. Grossniklaus, and M. H. Scholl, “Editorial: survey and experimental analysis of event detection techniques for twitter,” *The Computer Journal*, vol. 60, no. 3, pp. 329–346, 2016.