1-1-2021

Toward Tweet-Mining Framework for Extracting Terrorist Attack-Related Information and Reporting

Farkhund Iqbal
Zayed University, farkhund.iqbal@zu.ac.ae

Rabia Batool
Zayed University

Benjamin C. M. Fung
McGill University

Saiqa Aleem
Zayed University

Ahmed Abbasi
Air University Islamabad

See next page for additional authors

Recommended Citation

Iqbal, Farkhund; Batool, Rabia; Fung, Benjamin C. M.; Aleem, Saiqa; Abbasi, Ahmed; and Javed, Abdul Rehman, "Toward Tweet-Mining Framework for Extracting Terrorist Attack-Related Information and Reporting" (2021). *All Works*. 4792.
https://zuscholars.zu.ac.ae/works/4792

Author First name, Last name, Institution
Farkhund Iqbal, Rabia Batool, Benjamin C. M. Fung, Saiqa Aleem, Ahmed Abbasi, and Abdul Rehman Javed

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Tweet-to-Act: Towards Tweet-Mining Framework for Extracting Terrorist Attack-related Information and Reporting

FARKHUND IQBAL[1], RABIA BATOOL[1], BENJAMIN C. M. FUNG[2], SAIQA ALEEM[1], AHMED ABBASI[3], ABDUL REHMAN JAVED[3]
[1]College of Technological Innovation, Zayed University, Abu Dhabi, UAE
[2]School of Information Studies, McGill University, Montreal, Canada
[3]Department of Cyber Security, Air University, Islamabad, Pakistan

Corresponding author: Farkhund.Iqbal@zu.ac.ae

ABSTRACT The widespread popularity of social networking is leading to the adoption of Twitter as an information dissemination tool. Existing research has shown that information dissemination over Twitter has a much broader reach than traditional media and can be used for effective post-incident measures. People use informal language on Twitter, including acronyms, misspelled words, synonyms, transliteration, and ambiguous terms. This makes incident-related information extraction a non-trivial task. However, this information can be valuable for public safety organizations that need to respond in an emergency. This paper proposes an early event-related information extraction and reporting framework that monitors Twitter streams, synthesizes event-specific information, e.g., a terrorist attack, and alerts law enforcement, emergency services, and media outlets. Specifically, the proposed framework, *Tweet-to-Act* (T2A), employs word embedding to transform tweets into a vector space model and then utilizes the Word Mover's Distance (WMD) to cluster tweets for the identification of incidents. To extract reliable and valuable information from a large dataset of short and informal tweets, the proposed framework employs sequence labeling with bidirectional Long Short-Term Memory based Recurrent Neural Networks (*bLSTM-RNN*). Extensive experimental results suggest that our proposed framework, T2A, outperforms other state-of-the-art methods that use vector space modeling and distance calculation techniques, e.g., Euclidean and Cosine distance. T2A achieves an accuracy of 96% and an F1-score of 86.2% on real-life datasets.

INDEX TERMS
Terrorist Attacks, News, Word Embedding, Word Mover's Distance, Recurrent Neural Network, Information Extraction, Bidirectional Long Short-Term Memory

I. INTRODUCTION

Over the last few years, social media usage as an information source has dramatically increased due to the influx of smart connected devices and ease of accessibility. A recent study [1]–[3] shows that more than 40% of the world's population uses social media to connect and share information, resulting in an exponential increase in the volume of data. Researchers from various scientific domains use this data in different applications such as terrorism uses several approaches to carry out its plans and action, especially using social media platforms such as Twitter that use new technologies. Twitter is one of the most extensively used social media platforms that provide accurate predictions of terrorist activities and also has been studied as an emerging news reporting platform. Due to its accessibility and short text-based approach, it has become a medium of information dissemination that can break and spread news faster than traditional news media outlets [4], [5]. Terrorism is such a significant threat to many governments and people. Hence, monitoring and analyzing this rich flow of user-generated content can provide valuable information.

In emergencies, people often use Twitter to break the news, exchange information with media content, mobilize and unite, and raise funds for victims [6]. In such situations, efficient and effective actions are crucial for damage control and containment; thus, collecting and analyzing this data can be very valuable [7]. The data obtained during an event can provide information about injured, dead, and missing people and the urgent needs of affected people [8]. Law enforcement and emergency services, e.g., firefighters, paramedics, and rescue teams, are interested in finding ways to quickly and easily locate and organize the time-critical information that can be used to examine the situation and lead to appropriate action response [9]. Traditional media can also utilize this information to break the news. Additionally, this data can be helpful to digital forensic experts for evidence collection and to reach the eyewitnesses of the incident [8], [10].

While the data obtained can serve several practical purposes, one of the significant challenges lies in processing this large, recurrent, and noisy Twitter data to extract meaningful information. The challenges of information extraction from Twitter are different from the challenges of traditional media because tweets have length constraints and contain lots of informal, irregular, abbreviated words with spelling and grammatical mistakes. Moreover, the meaningless messages and rumors on Twitter are key factors affecting the overall performance and accuracy of the framework. Hence, there is a pressing need for an automatic framework that can *semantically* process noisy and recurrent data on Twitter and extract all the valuable information to generate concise reports about the event without any human intervention.

In this paper, we propose a novel framework, named *Tweet-to-Act* (*T2A*), that monitors the Twitter stream and, in case of a terror attack, automatically alerts media, emergency services, and law enforcement with concise and accurate information about the incident for a rapid response. The information contains the location and time of the attack, number of deaths and injuries due to attack, and potential involved persons and organizations. This information plays a vital role in assessing the severity of the situation and helps first responders to provide immediate help to the impacted citizens. Figure 1 shows the information flow of our proposed framework.

The main aim is to propose a framework for detecting terrorist attacks committed by terrorists by analyzing terrorist attack-related tweets from Twitter and reporting framework that monitors Twitter streams and synthesizes event-specific information. There are many people exposed daily to different forms of terrorist threats on social media platforms, which makes the early identification of these terrorist attack-related tweets paramount. Twitter is not just a platform for broadcasting information but an informative interaction. In order to develop a robust security framework to prevent this attack, people have now adopted sophisticated mechanisms with the help of various modern technologies.

The proposed framework T2A employs filtering and *agglomerative clustering* to similar group tweets from the incoming Twitter stream. tf-idf is the most used feature vector representation method for text; however, its design principle is based on the *bag-of-words* (BoW) model, which does not capture the position and semantics of words. To overcome the limitations of tf-idf, we use *word embedding*, a semantic and syntactic rich representation of words that captures the whole context of the word and also considers the surrounding words in a given document. To group similar tweets we employ a novel distance function called *Word Mover's Distance* (*WMD*) [11], which overcomes the synonym problem. Unlike other distance measures, WMD measures the distance between two documents in a meaningful way regardless of any commonality between words. Experimental results show that the use of WMD with *fastText* embedding improves the accuracy of the framework as compared to traditional methods.

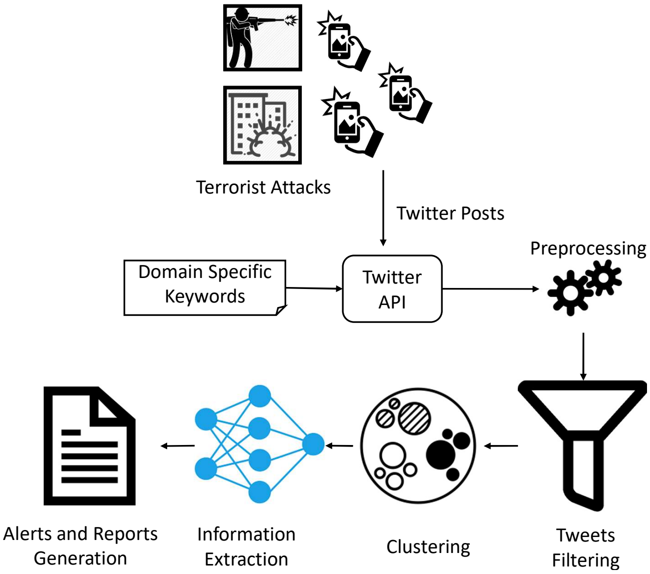


FIGURE 1: Proposed workflow of the terrorist attack identification and reporting framework

Named entity extraction is a sub-task of information extraction that helps to organize information in a structured way. It is a sequence labeling task where each token of a sentence is labeled with an entity type. *Hidden Markov Model* (*HMM*) and *Conditional Random Fields* (*CRF*) are well-known algorithms for sequence labeling tasks, but they do not consider semantics and long dependencies of a word on surrounding words when assigning a label. Feature selection and tuning by domain experts is another challenging task for HMM and CRF. To address these problems, we use a *bLSTM recurrent neural network*, which automatically learns the features without any human intervention and uses the previous and next state of the sequence to extract information. Our proposed bLSTM-based framework can achieve an accuracy of up to 96% and an F1-score of 86.2% when extracting attack-related information such as the location of the attack, number of injuries, and number of deaths.

In this paper, we make the following key contributions:

- Propose an event identification framework to identify terrorist attacks from a heterogeneous Twitter filtered stream and extract attack-related important details. Unlike traditional keyword-based approaches, our framework automatically generates embeddings that capture the semantics and importance of the tweets in a vector space.

- Propose a clustering method for the identification of incidents that employs word embedding and *Word Mover's Distance (WMD)* for early event identification and results in up to 2.5 times improvement in clustering performance in terms of silhouette score over state-of-the-art clustering approaches.
- Propose a method to extract event-related concise information from tweets using neural networks that automatically learn features without human intervention.
- Evaluate the proposed framework on real-world datasets and demonstrate the efficacy of the solutions against state-of-the-art text representation, clustering, and information extraction methods in terms of *silhouette score*, *adjusted mutual information score*, *accuracy*, and *F1-score*.
- Proposed framework outperforms other state-of-the-art methods that use vector space modeling and distance calculation techniques, e.g., Euclidean and Cosine distance. T2A achieves an accuracy of 96% and an F1-score of 86.2% on real-life datasets.

The rest of the paper is organized as follows: Section II discusses the closely related work. Section III discusses the proposed framework and its components in detail. Experiment settings and results of the proposed framework are discussed in Section IV. Section V concludes the work.

II. RELATED WORK

Due to the popularity of Twitter as an information channel, researchers are taking an interest in event extraction from Twitter. In recent years several methods have been proposed to extract new or recurrent events from Twitter [12]–[15]. For event identification problems, researchers have either focused on the general event: new emerging events, breaking news, and general topics that are discussed by a large number of users on Twitter [12], [16], [17], or specific events: known or planned events [18]–[20].

With other types of events, terrorism and crime identification using tracking and analysis of the Twitter stream have been a trending research area. Many researchers have analyzed the patterns of activities on social media during a terror attack. Ishengoma [6] analyzed the specific usage of online social networks at the time of terrorist attacks in developing countries. For this purpose, they used different metrics such as the number of tweets, user demographics, geolocation, and gender, and they also defined new metrics: reach and impression. Hughes and Palen [21] studied the use of Twitter during an emergency and national security events. Their study found that tweets contain much information that can be used for emergency management and response by authorities during such events. Gupta et al. [22] performed content and activity analysis on Twitter after the bomb blasts in Bombay to understand the dynamics and activities of online social media during a crisis. They highlighted that Twitter was being used for sharing information during such events instead of expressing personal opinions. They also discussed the spread of rumors during the blast. Moreover, Goolsby [23] also mentioned that in critical situations, like state terrorism, Twitter can be used as a source of information. In this paper, we are focusing on terrorism and crime events that are not known in advance. Compared to the discussed work, we are not just analyzing the statistics of tweets but proposing a framework that can automatically detect a terror attack and extract useful attack-related information from a filtered Twitter stream.

Some researchers have also worked on crime or terror attack detection. Amato et al. [24] proposed a framework to detect malicious actions, e.g., cyber-intrusion or terrorist activity in Twitter, that processes tweets related to social events and raises an alert in case there is any detected anomaly. Another framework proposed by Li et al. [25] detected crime and disaster events using tweets. They used Twitter-specific features like URL, hashtags, and mentions; domain-specific features like time, location, and user; and used to classify and rank multiple tweets. They also extracted spatial and temporal patterns of the event. Moreover, Marivate et al. [26] built a labeling framework for social security and crime incidents on social media. They extracted features from the text, user data, and the social network formed by user mentions building a classifier for data labeling. Alkhatib et al. [27] proposed a social media-based framework for incidents and events monitoring in smart cities using text classification and named entity recognition techniques. Our method monitors social media and extracts event-related useful information from tweets to be used by different organizations.

Meladianos et al. [28] proposed a method to detect sub-events and summarize any terror attack event using the Twitter stream. Sub-events are detected by monitoring the edge weights of graphs, and tweets for summarization are chosen using the greedy algorithm. Recently, Subramaniyaswamy et al. [29] performed sentiment analysis on real-time social media data to assess the public security threat used by law enforcement and intelligent purpose. By applying lexicon-based sentiment analysis on publicly available Twitter data, they measured the event's severity to assess the threat level. Another study by Harb et al. [30] applied deep learning techniques on tweets to classify emotions related to terrorism in terms of the emotional shift, emotions according to age and gender, and emotional reaction according to the closeness of the event and number of victims. Similarly, Laylavi et al. [31] proposed a novel method for detecting event-specific and informative tweets that could be valuable for emergency response by filtering and data-cleaning techniques and then applied a scoring method to measure the relatedness of a tweet to an event.

We propose an end-to-end framework that first monitors the Twitter stream to detect any terrorist attack and then extracts valuable information by processing tweets. To process Twitter data for event identification, most of the researchers used the traditional data representation method. tf-idf measures the importance of the word to a document, but it ignores the temporal order of words and the semantic and syntactic features. Our work analyzes the recently proposed word embedding for data representation. We use word mover distance instead of cosine and Euclidean distance in measuring the text-similarity for the early identification of terrorist attacks. So when the event is identified, we employ neural networks to extract event-related information that can keep the whole context into account. So the proposed approach outperforms other state-of-art approaches by using word distance instead of cosine and Euclidean distance.

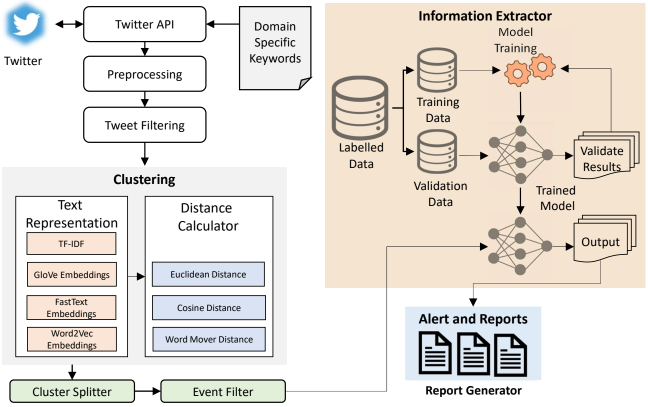


FIGURE 2: Architecture of the proposed framework Tweet-to-Act (T2A)

III. PROPOSED METHOD

The objective of this work is to design and implement a novel automatic terror attack detection framework, called *Tweet-to-Act* (*T2A*), that monitors the Twitter stream and, in case of a terror attack, extracts all related information shared on Twitter to alert law enforcement, emergency services, and media. The proposed approach is comprised of different phases: 1) Twitter API, 2) pre-processing, 3) tweet filtering, 4) clustering, 5) cluster splitter, 6) event identifier, 7) information extractor, 8) alerts and reports. Furthermore, the main problem can be divided into two sub-problems: terror attack detection and report generation. Figure 2 shows the architecture of the proposed framework T2A. There are two primary components in this architecture: clustering and information extraction. After preprocessing and filtering, the clustering engine uses word embedding and WMD to group similar tweets related to the terror attack. When the event is detected, the report generator uses the trained models to extract all the attack-related information without any human intervention automatically. This section describes each component in detail.

A. TWITTER API

For streaming real-time tweets, Twitter offers two APIs with varying numbers of filters and filtering capabilities. [32] Standard streaming API is free of cost but has a rate limit and allows only a single filtering rule per connection. Tweets returned by a search query are incomplete as the number of tweets must satisfy the limit imposed by Twitter. A paid enterprise API can filter real-time Twitter firehose using PowerTrack filtering language and facilitates multiple filtering rules per connection. Tweets can be filtered out based on various attributes, e.g., keywords, geolocation, language, etc. In our experiments, we set "English" as a default language. The user can enter keywords related to terror attacks such as blasts and shooting to retrieve the latest relevant tweets. Tweets collected in this method highly depend on the comprehensiveness of the search keywords. The framework starts with the primary keywords and auto-updates them along with event development from relevant tweets as described in [33]. Periodic timers are set to pass the collected stream of data to the event detection module every fifteen minutes [34].

B. PREPROCESSING

Data collected from social media is often noisy and heterogeneous. The preprocessing step removes all the mentions, URLs, special characters and stops tweets from making the Twitter stream ready for analysis. For the hashtags, it breaks down the complete hashtag into segments as segmented hashtags positively impact the data clustering [35]. Some hashtags are written using camel case, e.g., "#PrayForBoston" and are accessible to segment as they have defined word boundaries, but for hashtags that do not use camel case such as "#prayforboston," a large vocabulary is required to find the longest string matches in the hashtag. The framework uses a vocabulary of almost 70,000 English words for this purpose.

C. TWEET FILTERING

In the Twitter stream, filtering non-event-related messages is one of the significant challenges to be solved. To filter non-event tweets, we applied the Naive Bayes classification algorithm proposed by Ilina et al. [36]. Tweets have length constraints and contain lots of informal, irregular, abbreviated words with spelling and grammatical mistakes. Noise and redundancy of Twitter data also affect the fundamental analysis and its outcomes.After identifying event-related tweets, the filtering process removes all tweets from the stream that do not contain any informative content. For this purpose, the tweet filter counts the number of non-stop words and removes all the tweets that contain fewer than three words after preprocessing. It also removes all retweets and duplicated tweets from the stream as these tweets only reproduce the content of other tweets and do not add additional information. After preprocessing and filtering, the Twitter data is passed to the clustering module.

D. CLUSTERING

Clustering is defined as finding groups of objects in the data such that the objects in a group are similar to one another and different from the objects of other groups. Text clustering uses natural language processing to categorize unstructured text. Text clustering algorithms are divided into several types, such as agglomerative clustering algorithms, partitioning algorithms, and standard parametric modeling-based methods [37]. We opt for agglomerative hierarchical clustering as it does not require pre-specify the number of clusters like partition clustering. Agglomerative hierarchical clustering starts by assigning each tweet as a singleton cluster and agglomerates pairs of clusters based on their similarity until all clusters merge into a single cluster that includes all the tweets. It generates a cluster hierarchy where the leaf nodes correspond to individual tweets, and the internal nodes correspond to the merged groups of clusters. Agglomerative hierarchical clustering is more informative than flat clustering and can detect sub-events associated with the main terrorist attack. Different steps of clustering are explained in the following sections.

1) Text Representation

The clustering algorithm's quality depends on the features used in the clustering, so feature selection and representation is a very critical process. The representation of a set of documents as vectors in a common vector space is known as *vector space modeling*. Each dimension of this vector represents a separate term. There are several methods to compute these vectors. *Term Frequency and Inverse Document Frequency (tf-idf)* is the most popular and widely used scheme to compute the weighted value of each term. It reflects the importance of a term to a document in the corpus. For Twitter streams, tweet t_i can be described as

$$[W_{i1}, W_{i2}, \ldots, W_{ij}, \ldots, W_{in}]$$

where W_{ij} is the tf-idf value of jth term in the n-dimensional vector space. This method takes each term as an independent value and does not consider the semantic relation between terms.

Word embedding has emerged as another popular representation of text documents that consider not only the frequency but context, syntactic, and semantic similarity as well as relations with other words. Each word is represented by a real-valued vector having tens or hundreds of dimensions instead of millions of dimensions in one-hot encoded vectors. It results in a dense representation in which similar words have a similar encoding capturing their meaning. By examining the adjacency of words in this space, word embedding models can complete analogies such as "Man is to woman as king is to queen." Figure 3 shows different analogies that can be solved by applying arithmetic operations on word embeddings.

Word embedding has shown a good generalization power for feature representation in many NLP tasks, e.g., named entity recognition, dependency parsing, text classification. In our framework we use and compare three different kinds of word embeddings, namely *Word2Vec*, *GloVe*, and *fastText*.

Word2Vec [38] is a statistical method for efficiently learning a standalone word embedding from a text corpus considering a set of word pairs $\{(w_{ik}, c_{jk})\}_k$ generated from a large text corpus, by allowing the target word w_i to range over the corpus and the context word c_j to range over a context window. It is a combination of two models: *Continuous bag of words (CBOW)* and *Skip-gram*. Both are shallow neural networks that map words to the target variable. The CBOW model learns the embedding by predicting the current word based on its context, whereas the skip-gram model learns by predicting the surrounding words given a current word. When trained on extensive data, Word2Vec can generate a compressed vector representation that captures the semantic for each word. The generated representation, which is also known as embedding, can be used for clustering and classification. Specifically, the process focuses on learning about words given their local context, resulting in an equal numerical representation for similar words. However, they do not utilize the statistics of the corpus because they are trained on the local context windows and do not consider the global co-occurrence counts [39].

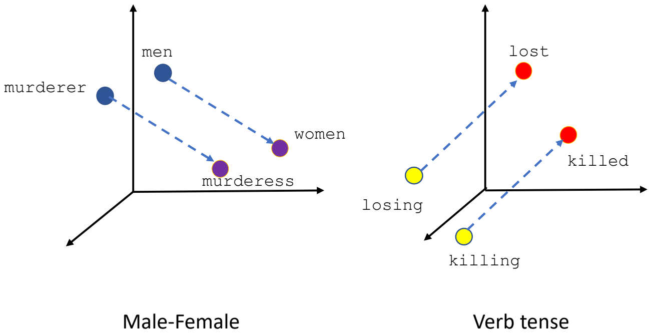


FIGURE 3: Analogy pairs relationships in word embedding space

Global Vectors for Word Representation (GloVe) [39] is an unsupervised learning algorithm for vector representation that can preserve semantic and syntactic regularities in the text. It works like Word2Vec but is a count-based model that trains on the word co-occurrence counts and thus makes efficient use of statistics. Compared to Word2Vec, the implementation of GloVe is easier to be parallelized, which is essential while training over a large dataset. The performance of GloVe and Word2Vec depends on the application domain and data.

fastText [40] is a library created by the Facebook Research Team for efficient learning of word representations. Compared to the Word2Vec model, it treats each word as composed of character n-grams so that the vector for a word is made of the sum of its character n-grams. It can generate better word embedding for rare and out of vocabulary words using its character n-gram.

2) Distance Calculator

In natural language processing applications such as classification and clustering, similarity/distance is an essential building block.In the early stage of an attack, there would be limited attack-related information. To tackle this issue, we have used different distance measures to cluster tweets for early detection of the incident and pass it to the neural networks to extract event-related concise information. There are several distance metrics for test vectors. *Euclidean distance* is a standard metric for geometrical problems. It measures the shortest distance among two documents using the Euclidean geometry. If the distance between two documents is zero, it means the documents are identical.

Cosine similarity measures the angle between two objects whose result ranges from 0 to 1. As the value of Θ increases, the value of cos Θ decreases, thus the less the similarity between two documents. The value 1 means two documents have the same orientation.

Euclidean distance and Cosine distance are known to do well in practice; however, they cannot capture the similarity when the same concept is written using different words. For example, consider these two sentences: *"a blast in Illinois killed two Chinese"* and: *"Report coming on the explosion in chicago where 2 people died."*. They both convey the same message but do not have any words in common, so the traditional distance calculation methods cannot find their similarity. The WMD method was introduced in 2015 by Matt Kusner et al. [11]. It adapts the *earth mover distance*. The WMD measures the similarity between two text documents in a meaningful way even if there is no commonality in the words of the two documents. Figure 4 illustrates how the word embedding model can present similar words close to each other in vector space. WMD leverages word embeddings and defines the distances between two documents as the optimal transport cost of moving all words from one document to another within the word embedding space. We opt to use WMD for two reasons. First, it can find the mean distance between the tweets written by different users in different linguistic styles. Second, it generates a good cluster representation in terms of tight and loose balance [41]. After measuring the distance/similarity between tweets, the framework merges the two most similar clusters at each iteration.

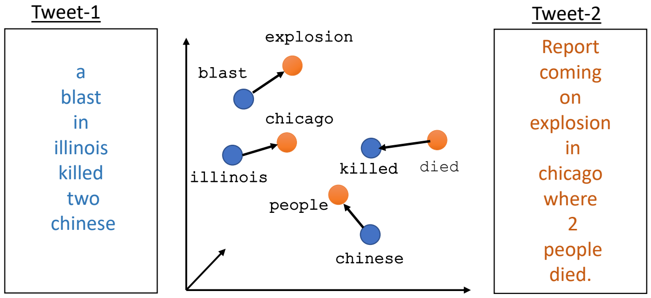


FIGURE 4: Semantically similar words in a word embedding space

E. CLUSTER SPLITTER

Hierarchical clustering does not require the number of clusters at the initialization phase. To obtain flat clusters, branches of the dendrogram can be cut at a specific level: a process referred to as tree cutting or dendrogram pruning. Cutting a hierarchy at a specific level gives a set of clusters while cutting at another level gives a different set of clusters; it depends on the application, data, and required granularity. A higher cutoff value results in looser clusters containing multiple events in the same cluster, while a lower cutoff value results in clean clusters but with a higher degree of event fragmentation. A common way to cut the hierarchy is using a constant height cutoff value, but, in the case of Twitter, there is a varying number of clusters and tweets, so the constant cutoff value does not perform well. We used a dynamic cutoff method [42] that depends on the data and tweets to be clustered. The dynamic cutoff is a flexible method to identify clusters from a complex hierarchical dendrogram based on its shape by decomposing and combining clusters iteratively. It first obtains a few large clusters by cutting the tree at the fixed height and then analyzes sub-cluster structure to split them recursively. To avoid over-splitting, it joins tiny clusters to their neighboring significant clusters.

F. EVENT IDENTIFIER

After clustering, all the identified clusters are passed to the event identifier. As Twitter data is noisy, it results in several small clusters with no helpful information or outliers. To filter outliers, the event identifier counts the number of tweets in each cluster. If the total number of tweets in a cluster is less than a threshold, it is discarded.We try an extensive range of threshold values and manually analyzed the clusters after cluster splitting and found that meaningful events clusters generally contain more than 50 tweets. Thus, setting the threshold at 50 would yield reasonably good results. Still, we leave this flexibility to the user. Each cluster has keywords related to terror attacks such as blast, shooting to retrieve the latest relevant tweets. At the end of the clustering process, each valid cluster represents an event with meaningful information passed to the next module to extract concise information about the attack.

In our experiments, we set "English" as a default language. The user can enter keywords related to terror attacks such as blasts, shooting to retrieve the latest relevant tweets.

G. INFORMATION EXTRACTOR

After identifying the terrorist attack events, the proposed framework processes all the tweets in the cluster to extract valuable information that the media can use, law enforcement, and rescue teams to analyze the situation and better plan for emergency response.

Recently, deep learning has obtained a very high performance in several NLP applications. *Recurrent neural network (RNN)* is an *artificial neural network* with loops that takes time and sequence into account and persists information of the previous state. RNN is flexible to use context, recognize sequential patterns in the presence of sequential distortions, and can be used with different data types and representations. These unique properties of RNN make it an optimal choice for sequence labeling [43].

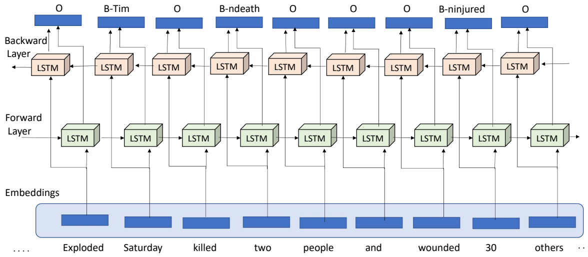


FIGURE 5: bLSTM network to extract attack related information

For an RNN network, given an input vector sequence x, denoted by

$$x = (x_1, ..., x_{t-1}, x_t, x_{t+1})$$

where x_t is the input at time step “t". The algorithm iterates over the following equations to update the hidden states of the network

$$s = (s_1, ..., s_{t-1}, s_t, s_{t+1})$$

and generate the outputs

$$y = (y_1, ..., y_{t-1}, y_t, y_{t+1})$$

where

$$s_t = f(Ux_t + Ws_{t-1} + b)$$

$$y_t = Vs_t + c$$

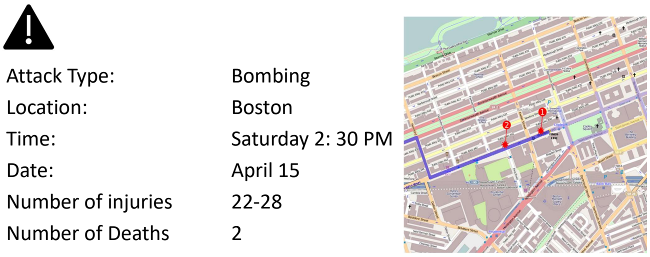
where the terms W, U, and V denote weight matrices connecting hidden to hidden, input to hidden, and hidden to output layers, respectively, and the terms b and c denote bias vectors. s_t is the hidden state at time step “t", which is calculated based on the previous hidden state and the input at the current step. The function f is a nonlinearity such as tanh or ReLU.

For a sequential labeling task, the LSTM model can consider an infinite amount of context and eliminate the problem of limited context that applies to any feed-forward model. *LSTM* networks [44] is a special kind of RNN architecture that can learn long-term dependencies. A bLSTM combines two LSTMs: one runs forward from "right to left," and one runs backward from "left to right." In bLSTM, the output layer takes information from the forward state as well as the backward state. This property of bLSTM makes it the best fit for our sequence labeling problem.

To extract valuable information from a Twitter post, we use sequence labeling, which assigns a class or label to each token in a given input sequence. The problem of sequence labeling can be defined as follows: For a given tweet regarding a specific terrorist attack, the task is to assign a label to each word of the tweet such that for an observation sequence w = (w^1, w^2, w^3,..., w^n), the output is a sequence of labels y = (y^1, y^2,y^3,...,y^n).

We leverage the power of bLSTM in which character embedding has been used to solve the problem of rare or unknown words. Each node in the input layer is connected with two separate hidden layers, one of which processes the input sequence of features forward, while the other processes it backward. Figure 5 shows the bLSTM network designed to extract attack-related information. To combine the output of the forward and the backward layer, there are options of concatenation, summation, multiplication, and average. We compared these options, as explained in the result section.

The forward and the backward LSTM hidden layers are fully connected to the input layer, and there are a total of 80 units in each layer. The output layer has a size equal to the number of tags to identify. The SoftMax activation function has been used for the output layer.



Attack Type:	Bombing
Location:	Boston
Time:	Saturday 2: 30 PM
Date:	April 15
Number of injuries	22-28
Number of Deaths	2

FIGURE 6: Sample of alert created for Boston attack

H. ALERTS AND REPORTS

After labeling tweets using the bLSTM model, the framework processes the extracted information to generate law enforcement and emergency services reports. These reports contain information such as the location, type and time of the attack, number of deaths, number of injuries. As the general public randomly writes tweets, sometimes multiple entities are given under the same label. For example, there could be a different number of deaths or injuries due to the attack mentioned by Twitter users. For such cases, the framework selects the range of those numbers from minimum to maximum in the report. For the location, it selects the most frequent location mentioned in the tweet cluster. Figure 6 shows the example of a report generated for the Boston Marathon bombing attack.

IV. EXPERIMENTS AND DISCUSSION

The framework uses *Gensim*, a Python library for automatically extracting semantic topics from documents [45] for feature representation. We evaluated different representation methods for tweets such as tf-idf, Word2Vec, GloVe, and fastText, which are all supported by Gensim. The details of these methods are explained earlier in Section III-D1. In our approach, we employ pre-trained word embeddings of Word2Vec, GloVe, and fastText for feature representation.

For clustering, *fastcluster API* [46], a Python library for hierarchical clustering, has been used; it offers faster clustering than *scipy* library [47]. We employ *Keras* [48]—a high-level API for the model representation of deep neural networks. Keras is highly used by *TensorFlow* [49] and PyTorch [50] because of its high-level abstraction, Python front-end, and

TABLE 1: Twitter event dataset (2012 -– 2016)

Event	Dates	Keywords
Boston Marathon bombing	Apr 15-16, 2013	boston attack, #prayforboston, bombing
Ferguson unrest	, Aug 9-26, 2014	#ferguson, violence
Ottawa shooting	Oct 22-24, 2014	ottawa, #ottawashooting, gunman shooting
Sydney siege	Dec 14-17, 2014	#sydneysiege, sydney, lindt,gunshot ,gunman, police gunfire
Charlie Hebdo shooting	Jan 7-14, 2015	#charliehebdo, #jesuischarlie, charlie hebdo, paris
Paris Attacks	Nov 13-24, 2015	#parisattacks, bataclan, paris
Brussels Airport explosion	Mar 22-30, 2016	brussels, airport, zaventem
Lahore blast	Mar 27-30, 2016	#lahoreblast, pakistan, lahore
Cyprus hijacked plane	Mar 29-30, 2016	#egyptair, hijacked, plane, cyprus, airport

support for CPUs and GPUs. The following section presents the details of our experimental setup, the dataset's choice, and the proposed framework's evaluation results.

A. DATASET

For the evaluation of terrorist attack detection, two different real-life event datasets are used. Dataset-A is taken from a Twitter event dataset (2012—2016) [51], [52]. This dataset contains data for 30 different events that happened from 2012 to 2016. Tweets are collected using the streaming API with a set of keywords. According to the Twitter terms of service, only tweet IDs are shared. We used *hydrator* [53] to collect Twitter data associated with these IDs. It automatically manages the rate limit of Twitter and returns tweets in JSON format. We chose the nine most important events related to terrorist attacks and crawled 1000 tweets for each event. Table 1 shows these selected events with their associated dates and the keywords used to collect those tweets.

Dataset-B, the most extensive Twitter event detection dataset, contains a collection of 120 million tweets, with relevance judgment of over 500 events [13]. The dataset contains Tweet IDs and their associated user IDs that can be used to crawl the actual tweets. Events are further split into eight categories: Business and Economy, Law and Politics, Science and Technology, Arts, Culture and Entertainments, Sports, Disasters and Accidents, Armed Conflicts and Attacks, and Miscellaneous.We filtered out 22 most discussed events on Twitter from Armed Conflicts and Attacks that are related to terrorist attacks and collected corresponding tweets using *hydrator*. Please note that this dataset was created in 2012, and we found that many tweets were either deleted or the user's account no longer exists. We found a total of 152,952 tweet IDs related to the previous events; however, only 72,662 (almost 47% of the original tweets) were successfully crawled. In order to reflect the Twitter live stream, we merged tweets for all events obtained from the datasets and fed them to the framework as a Twitter stream collected by Twitter API (as shown earlier in FIGURE 2).

In order to measure better accuracy of model and to extract the concise information extraction module, we used Kaggle feature engineered corpus annotated with *Inside–outside—beginning* (*IOB*) and *POS* tags for *Named Entity Recognition* [54]. IOB is a tagging method representing inside, outside, and the beginning of a chunk in the text. There is a total of 47,959 sentences containing 35,178 unique worlds with 17 different tags. We modified 1,668 sentences by adding extra tags for the number of deaths and the number of injured.

B. CLUSTERING

There are different linkage methods used for clustering, e.g., single, complete, average. In a single linkage, the distance between two clusters is the distance between their two closest objects. Mathematically, this can be represented as Eq. 1:

$$D(X,Y) = min(d(x,y)) : x \in X, y \in Y \tag{1}$$

The single linkage method controls only the nearest neighbor similarity and results in long thin clusters. On the other hand, in the complete linkage, also called the farthest neighbor, the distance between two clusters is the distance between the two most dissimilar objects. In this method, a pair of clusters is chosen whose merge results in the smallest diameter. Mathematically, this can be represented as Eq. 2:

$$D(X,Y) = max(d(x,y)) : x \in X, y \in Y \tag{2}$$

This linkage method is susceptible to outliers. In the average linkage method, the distance between two clusters is measured as the average distance between all pairs of objects containing objects from each group. Mathematically, this can be depicted as Eq. 3:

$$D(X,Y) = \frac{T_{XY}}{(N_X * N_Y)} \tag{3}$$

where T_{XY} is the sum of all pairwise distances between cluster X and cluster Y. N_{X} and N_{Y} are the number of objects in clusters X and Y, respectively.

We analyzed different linkage methods for agglomerative hierarchical clustering and found that the average linkage method outperforms single and complete linkage methods for Twitter data. Figure 7 shows the silhouette score with different linkage methods on dataset-A and dataset-B. The single linkage method outperformed complete and average linkage methods on dataset-A, but it did not perform well on dataset B, where the average linkage method showed the highest performance. Similarly, in FIGURE 8, it can be seen that the average linkage method outperformed complete and single linkage methods for both datasets. The single linkage method on dataset-A scored a very low Adjusted Mutual Information Score (AMIS). Hence it is not visible in FIGURE 8. Therefore, in the proposed framework, we chose the average linkage method for the clustering module.

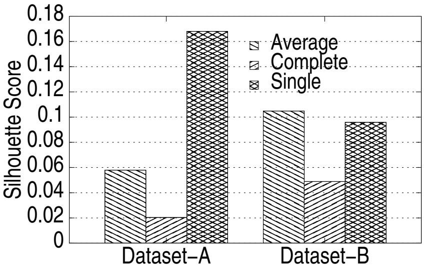


FIGURE 7: Silhouette score for dataset-A and dataset-B

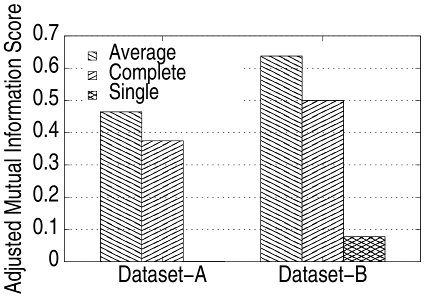


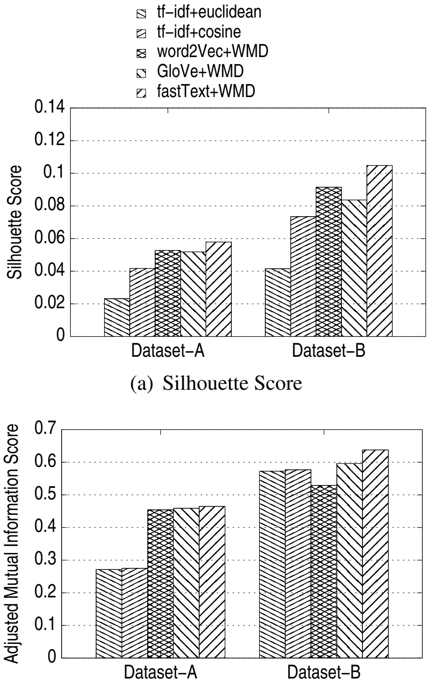
FIGURE 8: Adjusted mutual information score (AMIS) for dataset-A and dataset-B

We employed two popular metrics: *silhouette score* [55], [56] and *adjusted mutual information score (AMIS)* [57] to evaluate clustering using different features representation and distance calculation techniques. Silhouette score describes the ratio between cluster coherence and separation with value varying between -1 and 1. The mathematical description of Silhouette score is shown below as Eq.4:

$$sil = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{4}$$

The dissimilarity average of member i to all other members of the same cluster is depicted as $a(i)$, while $b(i)$ represents the dissimilarity average of member i to all its members closest cluster. Figure 9(a) shows the comparison of different feature representation and distance calculation methods for dataset-A and dataset-B. Amongst all the feature representation and distance calculation methods on both datasets, fastText combined with WMD has the highest silhouette score, closely followed by Word2Vec with WMD. Out of the five methods, word embedding with WMD has a higher Silhouette score as compared to state-of-art of-idf with euclidean and cosine distance. Euclidean distance with tf-idf representation resulted in the lowest silhouette score for the hierarchical clustering of Twitter data.

We took Euclidean distance with tf-idf as a baseline and compared it to other methods for the clustering performance. For dataset-A, the tf-idf+cosine method is 1.80 times better, Word2Vec+WMD is 2.28 times better, GloVe+WMD is 2.24 times better, whereas fastText+WMD showed 2.50 times score improvement than the baseline method. For dataset-B, to-idf+cosine is 1.77 times better, Word2Vec+WMD is 2.20 times better, GloVe+WMD is 2.02 times better, and fastText+WMD is 2.53 times better than the baseline method, which is by the results obtained on dataset-A.



(a) Silhouette Score

(b) Adjusted Mutual Information Score

FIGURE 9: Adjusted Mutual Information Score and Silhouette Score for Dataset A and Dataset B

The adjusted mutual information score is a measure of the similarity between two labels of the same data. The Mutual Information [57] between cluster U and V is given as Eq. 5:

$$MI(U,V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} \log \frac{N|U_i \cap V_j|}{|U_i||V_j|} \tag{5}$$

where $|U_{\mathrm{i}}|$ is the number of the samples in cluster U_{i}, and $|V_{\mathrm{j}}|$ is the number of the samples in cluster V_{j}. Adjusted Mutual Information Score (AMIS) is an adjustment of the Mutual Information Score (MIS) that corrects the effect of the agreement due to chance between clustering [58] as shown in Eq. 6.

$$AMI(U,V) = \frac{[MI(U,V) - E(MI(U,V))]}{[max(H(U),H(V)) - E(MI(U,V))]} \tag{6}$$

FIGURE 9(b) shows AMIS using different feature representation and distance calculation methods for dataset-A and dataset-B. It can be seen that the use of word embedding with WMD results in a higher AMIS as compared to the traditional approach of tf-idf with Euclidean and cosine distances. Amongst different word embeddings, the fastText resulted in the highest AMIS, closely followed by GloVe and Word2Vec. FIGURE 9(b) is also by the previous results, where the highest AMIS were observed with fastText and WMD for dataset-A and dataset-B. We used to-idf+euclidean as our baseline method and compared its performance with other approaches. For dataset-A, we found tf-idf+cosine to be 1.01 times better, Word2Vec+WMD to be 1.67 times better, GloVe+WMD to be 1.69 times better, and fastText+WMD to show 1.71 times improvement over the baseline. For dataset-B, the to-idf+cosine method showed 1.01 times, Word2Vec+WMD showed 0.92 times, and GloVe+WMD showed 1.04 times better performance fastText+WMD achieved 1.11 times better performance than the baseline. This indicates the viability of our proposed approach: it achieved up to 2.5 times improvement in clustering performance over state-of-the-art clustering approaches.

The proposed framework detected all the events with little fragmentation for dataset-A, which was resolved by applying the merging process after information extraction. If the generated reports were the same for more than one cluster, the framework merged the data and generated a single alert. For dataset-B, 5 out of 22 events had fewer than 50 tweets, and the count further decreased after filtering out duplicates and non-informative information. In a real-time framework, those events are likely to be detected by a subsequent cycle of tweets collection. Out of the remaining 17 events, 14 were successfully detected by the framework. Analysis of tweets related to undetected events revealed that those events were merged with other events due to the similarity between tweets. For example, "Damascus bomb kills at least 1" and "Syria Airstrikes Kill 8 In Damascus" are two tweets from different events. Similarly, "Gunmen killed 25 worshippers. JTT confirms blast in Maiduguri" and "News: Gunmen kill 20 at mosque in northern Nigeria" are two tweets from different events. Our framework detected these events and their associated tweets as a single event due to the similar nature of these events.

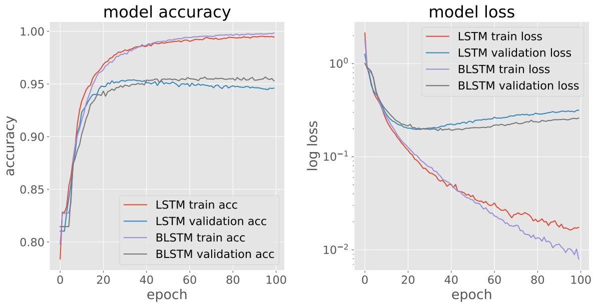


FIGURE 10: Training history of LSTM and bLSTM networks

C. INFORMATION EXTRACTION

TABLE 2: Accuracy of LSTM, and bLSTM

Model	Accuracy
LSTM	95.73% (+/- 0.31%)
bLSTM	96.27% (+/- 0.39%)

For information extraction, we used bLSTM networks trained using the aforementioned labeled data. Character embedding has been used in bLSTM networks, which helps to process unseen and out-of-vocabulary words. Data is split into three types: 1)*training data* — the actual dataset that is used to train the model, 2) *validation data* — sample of data used for an unbiased evaluation of the model while tuning model hyperparameters and 3) *test data* — sample of data used to evaluate the final model trained on the training dataset. 20% of the data is used as a test dataset; the remaining training dataset is further split into a training and validation dataset. The model is then iteratively trained and validated on these different datasets for *cross-validation*. We compared LSTM and bLSTM networks, and their training history graphs are shown in Figure 10. The graph on the left shows the accuracy of training and validation data, while the second graph on the right depicts the loss of the models on training and validation data with each epoch. It can be seen in the figure that bLSTM took more time to get better accuracy but has higher accuracy and lower loss compared to LSTM. This is due to the ability of bLSTM to interpret the long-range context of the sentence. After a certain number of epochs, we noticed *overfitting* of the model, i.e., instead of learning the general distribution of the data, the model just started to memorize the training data. This can be seen in Figure 10 as the loss keeps decreasing on the training data; however, after a certain threshold, it starts to increase. To avoid overfitting for the training data of both models, we used early-stopping of the training when values of loss function on the validation dataset were stabilized. We also used regularization of neural networks using *dropout* in which the model randomly selected the neurons that were dropped according to the assigned probability.

We performed 5-fold cross-validation for comparison. While LSTM networks consider only the previous state, bLSTM networks also consider the forward and backward states of the sequence. Table 2 shows the evaluation results of these models on the test dataset with ±standard deviation across five repetitions of the 5-fold cross-validation. In the case of bLSTM, the backward layer adds additional knowledge in model training, resulting in an accuracy gain of 96.27%. We further used *Precision*, *Recall*, and *F1-score* metrics to evaluate both models. *Precision* calculates how many predicted positives are actual positives, while *Recall* counts how many of the actual positives are correctly predicted by the model. *F1-score* is the harmonic mean of the Precision and Recall. These metrics are also calculated for each extracted label, as shown in Table 3. The framework extracted seven labels from the data. `ndeath` and `ninjured` are the numbers of deaths and number of injured, respectively. `org` and `per` represent possible involved organizations and persons in the attack. `gpe` is geopolitical, while `geo` indicates the location. `tim` is a time indicator of the attack. In the table, the higher values are written in bold. For `ninjured` and `gpe`, the LSTM network resulted in higher recall as compared to bLSTM. For `ninjured` this higher recall led to a higher F1-score of 0.86, as compared to 0.84 for bLSTM. In the case of `gpe`, precision is low, which

TABLE 3: Precision, Recall, F1-score, and Micro avg for LSTM and bLSTM networks

	LSTM			bLSTM		
Labels	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1- score**
ndeath	0.94	0.90	0.92	**0.95**	**0.99**	**0.97**
org	0.42	0.60	0.50	**0.62**	**0.58**	**0.58**
ninjured	0.84	**0.87**	**0.86**	**0.85**	0.83	0.84
gpe	0.97	**0.93**	0.95	**0.99**	0.91	0.95
tim	0.79	0.76	0.78	**0.80**	**0.80**	**0.81**
per	**0.75**	0.24	0.36	0.41	**0.44**	**0.42**
geo	0.72	0.85	0.78	**0.79**	**0.90**	**0.84**
micro avg	0.79	0.83	0.81	**0.85**	**0.87**	**0.86**
	F1-score = 81.2%			F1-score = **86.2%**		

TABLE 4: Merge Mode evaluation for bLSTM

Mode	**Accuracy**
Summation	96.27% (+/- 0.39%)
Concatenation	96.07% (+/- 0.15%)
Average	96.23% (+/- 0.29%)
Multiplication	95.80% (+/- 0.18%)

resulted in an F1-score equal to bLSTM. All the other labels resulted in higher recall values with the bLSTM network. In the case of precision, we can see that only the label got a higher score in the LSTM network, but due to lower recall, the F1-score remained lower than bLSTM. All other labels showed equal or higher precision with bLSTM. `ndeath` got an F1-score of 0.97 in bLSTM. In a multiclass classification problem with imbalanced class distribution, micro-average is preferable as it sums up the true individual positives, false positives, and false negatives of the framework for different classes before applying them to get the scores.We observed that bLSTM showed higher micro-average scores and a higher overall F1-score in comparison to the state-of-art LSTM network.

We also evaluated different merge modes of bLSTM. A merge mode is defined as combining forward and backward outputs to the next layer in bLSTM. There are four possible modes: summation, multiplication, concatenation, and average. Table 4 depicts the details of these modes. We can see that *summation* mode has the highest accuracy, i.e., 96.27% (+/- 0.39%), which is closely followed by *average*, which has an accuracy of 96.23% (+/- 0.29%). *Multiplication* has the lowest accuracy, i.e., 95.80% +/-0.18%. Based on these results, we selected *summation* mode for our model in the proposed framework.

V. CONCLUSION

In this paper, we presented a framework to extract terrorist attack-related information using the Twitter stream automatically and to generate alerts with concise information for media, emergency services, and law enforcement. The performance of clustering and event identification has been improved by filtering noisy data and applying semantic and syntactic similarity measures. We applied fastText word representation and WMD to group semantically-related tweets, which resulted in an up to 2.5 times improvement in clustering performance for detecting terrorist attacks over state-of-the-art clustering approaches. So the proposed framework, Tweet-to-Act (T2A), outperforms other state-of-the-art methods that use vector space modeling and distance calculation techniques for clustering methods to detect terrorist attacks.

For information extraction about an identified attack, we used a 2-layer bLSTM network, with one layer for the previous state and the other for the next stage of the sequence. Using the bLSTM with summation as the merge mode, the framework achieves 96% accuracy and an 86.2% F1-score. Media can use the generated reports and alerts to break the news, while law enforcement can use it to decide the rapid response for damage containment and public security. For future work, we aim to integrate other forms of input data, such as images and videos posted on Twitter, to extract more valuable and relevant information about incidents.

ACKNOWLEDGEMENT

This study is supported by Research Cluster Award (activity code: R17082) and Research Incentive Fund (activity code: R18055), Zayed University, United Arab Emirates.

REFERENCES

[1] M. A. Gilbert, “Strengthening your social media marketing with live streaming video,” in Smart Technologies and Innovation for a Sustainable Future, pp. 357–365, Springer, 2019.

[2] A. Abbasi, A. R. Javed, C. Chakraborty, J. Nebhen, W. Zehra, and Z. Jalil, “Elstream: An ensemble learning approach for concept drift detection in dynamic social big data stream learning,” IEEE Access, vol. 9, pp. 66408–66419, 2021.

[3] W. Ahmed, F. Shahzad, A. R. Javed, F. Iqbal, and L. Ali, “Whatsapp network forensics: Discovering the ip addresses of suspects,” in 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–7, IEEE, 2021.

[4] M. Cheong and V. C. Lee, “A microblogging-based approach to terrorism informatics: Exploration and chronicling civilian sentiment and response to terrorism events via twitter,” Information Systems Frontiers, vol. 13, no. 1, pp. 45–59, 2011.

[5] 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, 2010.

[6] F. R. Ishengoma, “Online social networks and terrorism 2.0 in developing countries,” CoRR, vol. abs/1410.0531, 2014.

[7] H. Chen and J. Xu, “Intelligence and security informatics,” Annual review of information science and technology, vol. 40, no. 1, pp. 229–289, 2006.

[8] K. Zahra, M. Imran, F. O. Ostermann, K. Boersma, and B. Tomaszewski, “Understanding eyewitness reports on twitter during disasters,” in ISCRAM, ISCRAM, 2018.

[9] S. Vieweg, C. Castillo, and M. Imran, “Integrating social media communications into the rapid assessment of sudden onset disasters,” in International Conference on Social Informatics, pp. 444–461, Springer, 2014.

[10] A. R. Javed and Z. Jalil, "Byte-level object identification for forensic investigation of digital images," in 2020 International Conference on Cyber Warfare and Security (ICCWS), pp. 1–4, IEEE, 2020.

[11] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in International Conference on Machine Learning, pp. 957–966, 2015.

[12] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," in Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics, pp. 181–189, Association for Computational Linguistics, 2010.

[13] A. J. McMinn, Y. Moshfeghi, and J. M. Jose, "Building a large-scale corpus for evaluating event detection on twitter," in Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pp. 409–418, ACM, 2013.

[14] D. Zhou, L. Chen, and Y. He, "A simple bayesian modelling approach to event extraction from twitter," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, pp. 700–705, 2014.

[15] F. Kunneman, A. van den Bosch, F. Grootjen, M. Otworowska, and J. Kwisthout, "Event detection in twitter: A machine-learning approach based on term pivoting," in Grootjen, F.; Otworowska, M.; Kwisthout, J.(ed.), Proceedings of the 26th Benelux Conference on Artificial Intelligence, pp. 65–72, Nijmegen, the Netherlands: Radboud University, 2014.

[16] 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, 2009.

[17] J. Weng and B.-S. Lee, "Event detection in twitter," in Fifth international AAAI conference on weblogs and social media, 2011.

[18] 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, 2010.

[19] E. Benson, A. Haghighi, and R. Barzilay, "Event discovery in social media feeds," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 389–398, Association for Computational Linguistics, 2011.

[20] 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, 2010.

[21] A. L. Hughes and L. Palen, "Twitter adoption and use in mass convergence and emergency events," International Journal of Emergency Management, vol. 6, no. 3-4, pp. 248–260, 2009.

[22] A. Gupta and P. Kumaraguru, "Twitter explodes with activity in mumbai blasts! a lifeline or an unmonitored daemon in the lurking," tech. rep., Indraprastha Institute of Information Technology, Delhi, 2012.

[23] R. Goolsby, "Lifting elephants: Twitter and blogging in global perspective," in Social computing and behavioral modeling, pp. 1–6, Springer, 2009.

[24] F. Amato, G. Cozzolino, A. Mazzeo, and S. Romano, "Detecting anomalies in twitter stream for public security issues," in 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), pp. 1–4, IEEE, 2016.

[25] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang, "Tedas: A twitter-based event detection and analysis system," in 2012 IEEE 28th International Conference on Data Engineering, pp. 1273–1276, IEEE, 2012.

[26] V. Marivate and P. Moiloa, "Catching crime: Detection of public safety incidents using social media," in 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), pp. 1–5, IEEE, 2016.

[27] M. Alkhatib, M. El Barachi, and K. Shaalan, "An arabic social media based framework for incidents and events monitoring in smart cities," Journal of Cleaner Production, vol. 220, pp. 771–785, 2019.

[28] P. Meladianos, C. Xypolopoulos, G. Nikolentzos, and M. Vazirgiannis, "An optimization approach for sub-event detection and summarization in twitter," in European Conference on Information Retrieval, pp. 481–493, Springer, 2018.

[29] V. Subramaniyaswamy, R. Logesh, M. Abejith, S. Umasankar, and A. Umamakeswari, "Sentiment analysis of tweets for estimating criticality and security of events," in Improving the Safety and Efficiency of Emergency Services: Emerging Tools and Technologies for First Responders, pp. 293–319, IGI Global, 2020.

[30] J. G. Harb and K. Becker, "Comparing emotional reactions to terrorism events on twitter," in Workshop on Big Social Data and Urban Computing, pp. 107–122, Springer, 2018.

[31] F. Laylavi, A. Rajabifard, and M. Kalantari, "Event relatedness assessment of twitter messages for emergency response," Information processing & management, vol. 53, no. 1, pp. 266–280, 2017.

[32] "Filter realtime tweets." https://developer.twitter.com/en/docs/tweets/filter-realtime/overview. Accessed: 2020-01-05.

[33] X. Zheng and A. Sun, "Collecting event-related tweets from twitter stream," Journal of the Association for Information Science and Technology, vol. 70, no. 2, pp. 176–186, 2019.

[34] G. Ifrim, B. Shi, and I. Brigadir, "Event detection in twitter using aggressive filtering and hierarchical tweet clustering.," in SNOW-DC@ WWW, pp. 33–40, 2014.

[35] D. Gromann and T. Declerck, "Hashtag processing for enhanced clustering of tweets.," in RANLP, pp. 277–283, 2017.

[36] E. Ilina, C. Hauff, I. Celik, F. Abel, and G.-J. Houben, "Social event detection on twitter," in International Conference on Web Engineering, pp. 169–176, Springer, 2012.

[37] C. C. Aggarwal and C. Zhai, "A survey of text clustering algorithms," in Mining text data, pp. 77–128, Springer, 2012.

[38] "word2vec." https://code.google.com/archive/p/word2vec/, 2013. Accessed: 2020-01-10.

[39] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543, 2014.

[40] F. Inc, "fasttext: Library for efficient text classification and representation learning." https://fasttext.cc/, 2016. Accessed: 2020-01-10.

[41] N. Franciscus, X. Ren, J. Wang, and B. Stantic, "Word mover's distance for agglomerative short text clustering," in Asian Conference on Intelligent Information and Database Systems, pp. 128–139, Springer, 2019.

[42] P. Langfelder, B. Zhang, and S. Horvath, "Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for r," Bioinformatics, vol. 24, no. 5, pp. 719–720, 2007.

[43] K. Kawakami, Supervised sequence labelling with recurrent neural networks. PhD thesis, Ph. D. thesis, Technical University of Munich, 2008.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Citeseer, 2010.

[46] "fastcluster 1.1.25." https://pypi.org/project/fastcluster/, 2012. Accessed: 2020-01-10.

[47] "Clustering package (scipy.cluster)." https://docs.scipy.org/doc/scipy/reference/cluster.html, 2018. Accessed: 2020-01-10.

[48] A. Gulli and S. Pal, Deep Learning with Keras. Packt Publishing Ltd, 2017.

[49] S. S. Girija, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," Software available from tensorflow. org, 2016.

[50] N. Ketkar, "Introduction to pytorch," in Deep learning with python, pp. 195–208, Springer, 2017.

[51] A. Zubiaga, "Twitter event datasets (2012-2016)." https://figshare.com/articles/Twitter_event_datasets_2012-2016_/5100460, 2018. Accessed: 2020-01-10.

[52] Z. Arkaitz, "A longitudinal assessment of the persistence of twitter datasets," Journal of the Association for Information Science and Technology, vol. 69, no. 8, pp. 974–984, 2018.

[53] "hydrator." https://github.com/DocNow/hydrator, 2018. Accessed: 2020-01-10.

[54] A. Walia, "Annotated corpus for named entity recognition." https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus, 2017. Accessed: 2020-01-10.

[55] L. Kaufman and P. J. Rousseeuw, Finding groups in data: an introduction to cluster analysis, vol. 344. John Wiley & Sons, 2009.

[56] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," Journal of computational and applied mathematics, vol. 20, pp. 53–65, 1987.

[57] "mutual_info_score." https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mutual_info_score.html, 2011. Accessed: 2020-01-10.

[58] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," Journal of Machine Learning Research, vol. 11, no. Oct, pp. 2837–2854, 2010.

FARKHUND IQBAL holds the position of Associate Professor and Director Advanced Cyber Forensics Research Laboratory in the College of Technological Innovation, Zayed University, The United Arab Emirates. He holds a Master (2005) and a Ph.D. degree (2011) from Concordia University, Canada. He is using machine learning and Big Data techniques for problem-solving in healthcare, cybersecurity, and cybercrime investigation in smart and safe city domain. He has published more than 80 papers in peer-reviewed high-ranked journals and conferences. He is an affiliate professor in the school of information studies, McGill University, Canada, and Adjunct Professor, Faculty of Business and IT, University of Ontario Institute of Technology, Canada. He is the recipient of several prestigious awards and research grants. He has served as a chair and TPC member of several IEEE/ACM conferences, guest editor of special issues, and reviewer of high-rank journals. He is a member of several professional organizations, including ACM and IEEE Digital society.

RABI BATOOL a researcher in the College of Technological Innovation at Zayed University. She received a Master's Degree in computer science from Kyung Hee University, South Korea. Her research interests are Data mining, Deep Neural Networks, Natural language Processing, and Information Extraction have published several research papers in these areas.

BENJAMIN C. M. FUNG is a Canadian Research Chair in Data Mining for Cybersecurity, an Associate Professor in the School of Information Studies at McGill University, and a Co-curator of Cybersecurity in the World Economic Forum (WEF). He received a Ph.D. degree in computing science from Simon Fraser University in 2007. He has over 120 refereed publications that span data mining, machine learning, privacy protection, cyber forensics, and building engineering research forums. His data mining works in crime investigation and authorship analysis have been reported by media worldwide. Dr. Fung is a licensed professional engineer in software engineering. He is a Senior Member of both ACM and IEEE.

SAIQA ALEEM received her M.S. degree in computer science from University of Central Punjab, Pakistan, in 2004, M.S. degree in information technology from UAEU, United Arab Emirates University, in 2013, and a Ph.D. degree in Electrical and Computer Engineering from the University of Western Ontario, Canada, in 2016. Currently, she is an assistant professor at Zayed University. She had many years of academic and industrial experience, holding various technical positions. She is Microsoft, CompTIA, and Cisco certified professional with MCSE, MCDBA, A+ and CCNA certifications. Her research interests are Game development process, cloud computing, software process assessment models, IoT, and social network analysis.

AHMAD ABBASI is with the Department of Cyber Security, Air University, Islamabad, Pakistan He is working with National Cybercrimes and Forensics Laboratory, Air University, Islamabad, Pakistan. He is doing his Master's degree in Data Science from Air University, Islamabad, Pakistan.

ABDUL REHMAN JAVED is a lecturer at the Department of Cyber Security, Air University, Islamabad, Pakistan. He worked with National Cybercrimes and Forensics Laboratory, Air University, Islamabad, Pakistan. He received his Master's degree in Computer Science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan. His current research interests include but are not limited to mobile and ubiquitous computing, data analysis, knowledge discovery, data mining, natural language processing, smart homes, and their applications in human activity analysis, human motion analysis, and e-health. He aims to contribute to interdisciplinary research of computer science and human-related disciplines. He has authored more than 40 peer-reviewed articles on cybersecurity, mobile computing, and digital forensics topics.

• • •