A Real-Time Emergency Event Detection and Location Prediction Framework for Twitter Streams

Sako, D.J.S.
Department of Computer Science
Rivers State University
Port Harcourt, Nigeria

Onuodu, F.E. & Eke, B.O.
Department of Computer Science
University of Port Harcourt
Rivers State, Nigeria

ABSTRACT
This paper presents a methodology that combines the spatio-temporal metadata of Twitter posts with the analysis of their contents to filter the stream data for emergency situation reporting and visualization. Twitter messages are monitored in real time for a number of small scale emergency events by streaming and analyzing the contents of the posts using Kafka-enabled technology and convolutional neural network deep learning classification models, respectively, to determine whether or not they are incident-related. All the incident-related posts are further classified into the emergency categories they belong to and then geocoded to determine their locations by considering tweet geotags (GPS location), tweet place, textual content for mentioned location and users' profile location as the main location-related elements. This results in a significantly higher rate of tweets with associated location information, and hence enables tweet location analysis and visualization for smaller events. As an application, we developed an incident detection, notification and reporting system that detects emergency events promptly and sends notification e-mails to the appropriate agencies for timely response.

KEYWORDS: *Incident Detection, Location Prediction. Classification, Twitter, Geo-tagging*

1. INTRODUCTION

Each society and its components (individuals, groups, organizations, communities, etc) are occasionally and usually in an unexpected way exposed to disasters, crisis/emergency situations. Whether an emergency situation is triggered by a natural phenomenon (e.g. storms, earthquakes, bushfire, flooding, etc.), or at the instigation of humans (terrorism, riots, communal conflicts, crimes, shootings, a technological failure or an industrial accident, widespread power outages, explosions, structure collapses, fire in residential buildings, etc.), the normal functioning of a society and its parts will be disturbed in a more or less serious manner (De Smet et al., 2011). With the advent of social media new means to collect information that contributes to situation awareness in an emergency situation have emerged (Schulz et al., 2016). The ubiquity of social media platforms presents an opportunity to harness developing information to improve situation awareness for management and response teams (Lagerstrom et al., 2016) during crisis or emergency. This is gained based on information about the environment within the volume of time and space affected by the crisis (Endsley, 1988).

The enormous amount of data coming from social media needs to be processed, classified and visualized on the fly in order to detect and monitor unplanned emergency events, e.g., accidents, attacks, etc. Analyzing and visualizing stream data has become necessary in emergency management and response as this will help track changes in stream data. Given this large volume of messages published by individuals on the social media (e.g Twitter), it is obvious that by combining the spatio-temporal attributes of these messages with the content posted, sensitive information about people and their locations can be identified (Kelly et al., 2017)
An incident is among the most unfavorable events that significantly affect the society and decrease the reliability of the system. The impact of an incident or disruption due to the incident could be minimized by implementing real-time intervention strategies. The effectiveness of the intervention measures depends on accurate real-time information about the location, duration, and impact of the incident. Knowing physical locations involved in Twitter helps us to understand what is happening in real life (Zheng et al., 2018).

An important characteristic of Twitter, a free social networking and microblogging service, is its real-time nature. The challenge of analyzing real time stream data lies in reducing the difference between the time of retrieval and the time of analysis. A delay in processing one of these tasks creates a cumulative gap between the analysis time and the retrieval time and leads to the stream getting disconnected.

Tweets are short messages that are restricted to 280 characters in length. However, a tweet is more than a short message. Tweets come bundled with a relatively rich set of meta data. Through the streaming API, subsets of public status descriptions can be retrieved based on user-defined criteria in JavaScript Object Notation (JSON) formatted data which is a lightweight and text-based data exchange format (Laylavi, etal., 2016). A tweet can be associated with a time and location, which is a set of latitude and longitude coordinates; each tweet has its post time, which is obtainable using a twitter search API and contained in the twitter feed. GPS data (geotags) are attached to a tweet sometimes, e.g. when a user is using a smartphone. However, considering that less than 2% of tweets are geotagged, finding location inference methods that can go beyond the geotagging capability. This is especially true in terms of emergency response, where spatial aspects of information play an important role in order to understanding the impact of an emergency situation or disaster, including where the damage is, where people need assistance and where help is available (Lingad , et al., 2013; Ao et al, 2014).

Table 1: Incident types and their description

Disaster/Crisis Type	Description
Damaged Infrastructure	Collapsed/Damaged buildings or roads, destroyed bridges, utilities/services interrupted (e.g falling electricity pole).
Fire	Building fire
Flood	floods caused by heavy rainfall
Traffic Accidents	Motor vehicle crash such as motorcycle accident, car accident, plane crash, etc.
Crime and Civil Disorder	shootings, cultism and terrorism, kidnapping, riot, protest, etc

Getting location information out of crisis tweets by will help emergency response in a clear and timely manner. To do this, we need to identify the correct location in order to send help to the right place. For each tweet multiple locations could be found inside its text and metadata; places mentioned within the tweet text, tweets geotagged with an exact location or Twitter Place with Geographical Positional System (GPS) information and user profile. GPS information of are considered the most accurate for tweet location. However, as reported by Cheng et al. (2010), only 0.42% tweets are geotagged, whereas Morstatter et al. (2013) reported that around 3.17% tweets are geotagged, making it difficult to extract non-geotagged tweets from an area during emergencies.

In order to overcome these challenges and extend the models trained using historical data to a real-time model, we propose a Kafka-supported deep learning based approach to classify tweets for signs of a variety of small scale emergency event types including damaged infrastructure, crimes, fire, accident and flood (Table 1) and inferring the location of the incident by combining the spatio-temporal attributes of these messages with the contents in real-time.

As a streaming platform, Kafka provides a highly scalable, fault tolerant, publish and subscribe pipeline of message queue architected as a distributed transaction log, making it highly valuable for enterprise infrastructures to process streaming data (source) in real time. In the simplest way there are three players in the Kafka ecosystem: producers, topics (run by brokers) and consumers (Dorpe, 2018). We create a decoupled twitter stream where we essentially divide the stream into two different modules. One module known as the "Producer", which is defined by specifying the Kafka Broker and the topic name, collects the data from the twitter stream, then saves it as logs (new messages are saved at the end of the log) into the Kafka Cluster (queue) without doing any processing. Another module known as the "Consumer" reads the logs (in the queue or Kafka cluster) and processes the data separately. This allowed us to process the raw data from twitter without worrying about the stream getting disconnected because now we have essentially decoupled the entire Twitter stream

To assign an incident category to a tweet, two data filtering/classification processes are performed using convolutional neural networks: (1) determine whether or not the tweet is incident-related; (2) if a tweet is incident-related, then which category it belongs to. Table 2 and 3 show sample tweets in these categories.

To geocode or infer the location of the tweet, we introduce a multi-elemental location inference method that predict the location of tweets by exploiting the geotagging and inherently attached data elements.

In the next section, we present existing work on emergency tweet classification and location estimation. Section 3 discusses the approaches and methods used in this study. We describe our dataset and report on the experimental results in section 4. In section 5 we present the emergency situation alert and reporting system developed using our methodology of emergency event detection and location estimation. Finally, we conclude with final remarks in section 6.

Table 2: Sample tweets for the incident/non-incident classification

Tweet	*Label*
Two workers feared trapped as two-storey building collapses #BuildingCollapse	*incident*
Success is no accident. it is hard work, perservance, learning, studying sacrifice, and most of all love of what you are doing	*non-incident (false positive)*
if you are making the trip to #cheltenhamfestival .... #accident on m6 southbound between junctions 8 and	*incident*
#reno Truck trailer catches fire in Reno http://t.co/k5FIJaNkJb	*incident*

Dreaming of this after a looooong day of Black Friday shopping #HoteletteNashville (photo cred: @catherinetrumanphoto)	*non-incident*
Homecoming Queen Killed on Way Home from the Prom by Flood Waters! #socialnews http://t.co/VmKexjTyG4	*incident*
My first staining attempt was a disaster https://t.co/buDmKE3nNf	*non-incident*

Table 3: Sample tweets for the incident type classification

Tweet	***Label***
Two workers feared trapped as two-storey building collapses #Berekusu, #BuildingCollapse, #EasternRegionGhana	*Infrastructure damage*
if you are making the trip to #cheltenhamfestival .... #accident on m6 southbound between junctions 8 and 7	*accident*
At least 50 Dead and Several Others Injured as Suicide Bomber attacks Mosque in Adamawa State. Read more @ www.celestreet.com #SuicideBomber #Mosque #AdamawaState #Mubi #MubiNorthLGA #MadinaMosque #BokoHaram #SuicideBombing #Terrorism	*Crime and civil disorder*
Homecoming Queen Killed on Way Home from the Prom by Flood Waters! #socialnews http://t.co/VmKexjTyG4	*Flood*
#reno Truck trailer catches fire in Reno http://t.co/k5FIJaNkJb	*Fire*

2. RELATED WORK

Spatio-temporal models have been investigated in a number of studies to find the location of a targeted event in different domains such as earthquakes (Sakaki et al., 2010; Sakaki et al., 2013; Yin et al., 2012), flooding events (Kelly, eta al 2017), fires (Vieweg etal., 2010), conflicts and political disasters (Dou et al., 2012; Ritter, et al, 2012; Robinson et al.,2013), societal events (Ciulla et al. 2012), political elections (Skoric et al. 2012), tourist spots (Oku et al. 2014) and public health (Paul and Dredze 2011), with some systems attempting to automatically detect events (Li et al., 2012).

Researchers have employed different machine learning, statistical, probability and natural language processing techniques to estimate the location from tweets (Ajao et al. 2015). Most works have considered the geographical references used in tweets to determine the location in the absence of geotagging. These geographical references are either “location indicative words” (LIWs) such as local dialectal terms (e.g. yinz) and place names (e.g. Portland) (Bo et al. 2012) or gazetteer terms.

Sakaki et al. (2010, 2013) investigated the real-time interaction of events such as earthquakes, in Twitter, and proposed an algorithm to monitor tweets and to detect a target event. They produced a probabilistic spatiotemporal model for the target event that can find the center and the trajectory of the event location. They considered each Twitter user as a sensor and applied Kalman filtering and particle filtering, which are widely used for location estimation in ubiquitous/pervasive computing.

Ao et al. (2014) proposed a method to detect the location where some events are occurring by analyzing only data from weibo information. In order to detect event-aware location, they modelled three kinds of location data of each weibo to tackle with detection challenge: (1) content-based location; (2) posting location; (3) registration location of author of weibo. After they obtained a set of event-aware locations, they use a hierarchical clustering to estimate the accurate event location. Their result showed that their method could improve the accuracy of event location estimation.

Singh et al. (2017) investigated the Twitter post in a flood related disaster and proposed an algorithm to identify victims asking for help. The developed system takes tweets as inputs and categorizes them into high or low priority tweets. User location of high priority tweets with no location information is predicted based on historical locations of the users using the Markov model.

Mahmud et al. (2014) presented a hierarchical ensemble algorithm for inferring the home location of twitter users at different granularities, including city, state, time zone or geographic region, using the content of users' tweets and their tweeting behavior. Their algorithm uses a variety of different features, leverages domain knowledge and combine an ensemble of statistical and heuristic classifiers to predict locations and makes use of a geographic gazetteer dictionary to identify place-name entities. They also analyzed movement variations of Twitter users, built a classifier to predict whether a user was travelling in a certain period of time and use that to further improve the location detection accuracy.

3. APPROACHES AND METHODS:

In this section, we present our approach for emergency events classification and location estimation using big data system like Apache Kafka and Deep Learning Convolutional Neural Network models. Figure 1 outlines the design and architecture of the proposed method. In general, the proposed framework consists of five main components: Kafka-based Data collection, filtering and processing, training and prediction model, geocoding/location inference and the web-based GUI for visualization and reporting.

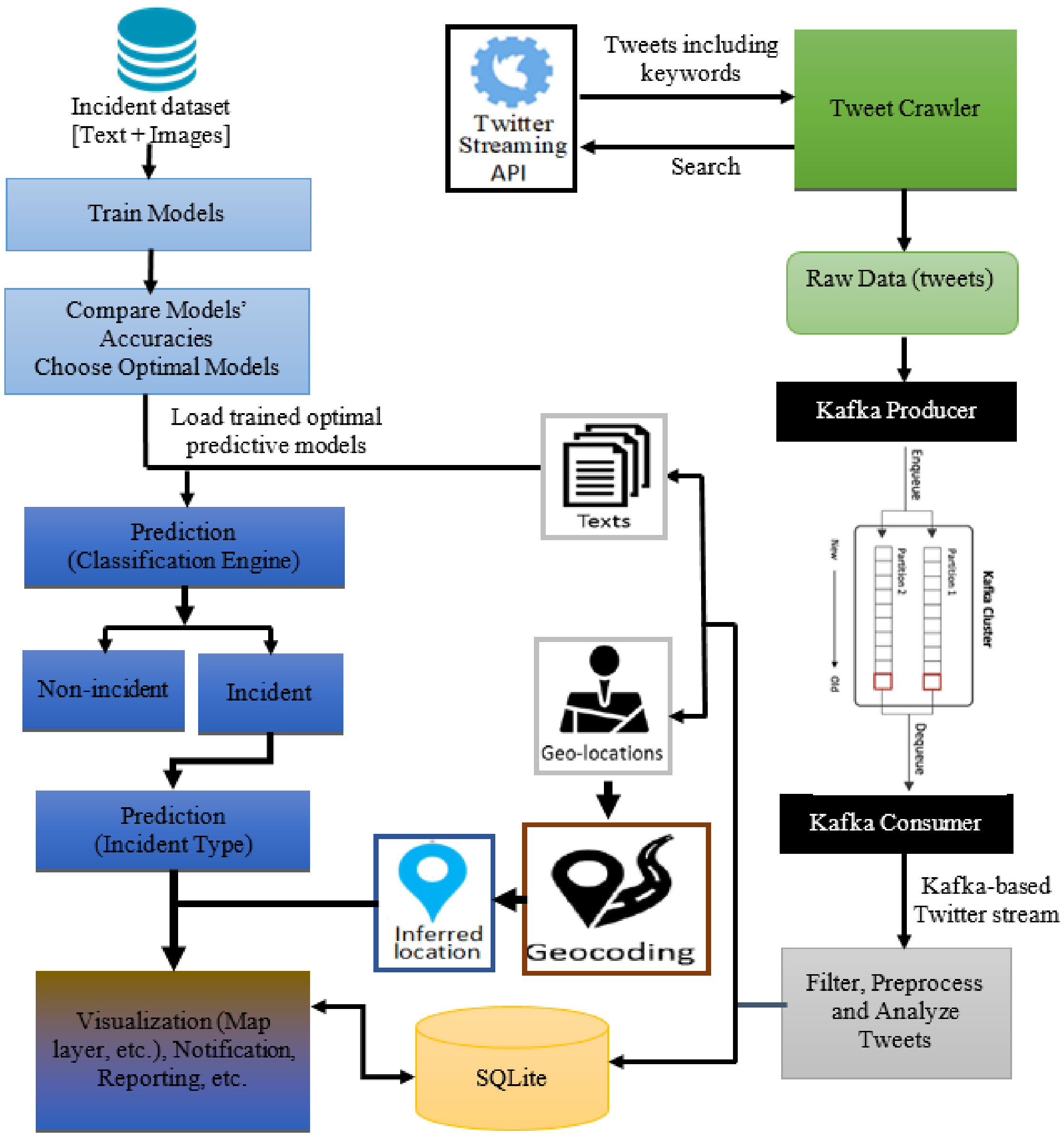

Figure 1: The framework for the implementation of the Proposed System

3.1 Twitter Data Collection:

In Twitter research, which can be generally characterized as data-driven, the Twitter Streaming API provides low latency access to subsets of real-time stream of public tweets. However, it does not support queries with location and keywords simultaneously, and it only gives access to 1% of the volume of tweets per second at that moment. The connection to Twitter was made through Python. To train our CNN models, we collected data from the twitter platform using a dictionary (collection) of keywords and hashtags queries and filter matching for twitter texts over a period of time to ensure that a representative sample of incident related tweets are collected.

We used Apache Kafka to streamline collection of real-time data from twitter stream to allow processing, classification, geocoding and analysis to be done on the fly. So instead of having a single Python module that collects, processes, classifies and saves everything into a JSON file, we create a decoupled twitter stream where we essentially divide the stream into two different modules (producer and consumer) to avoid the stream getting disconnected due to delay gap between streaming and processing.

3.2 Data Filtering and Preprocessing

The streamed data is filtered and segmented to extract available and needed information. For each of the retrieved messages, some of the basic information extracted, where they are present in the message, include: tweet text, user, tweet id, tweet time, geolocation information (geo tags, place, mentioned in text, and location field in user profiles,), etc.

A number of steps are used to clean the tweets for this study: If the tweet contained "RT", then that tweet was deleted, especially in real time analysis, as this was not originally created by the sender, and it did not qualify for our analysis. We discarded non-English tweets, deleted all non-ASCII characters, hashtag signs (#) from the beginning of all hashtag words, and all the punctuation marks and numbers. The stop words were removed, as they do not convey any meaningful information. Any unwanted multiple dots were removed, and multiple spaces were merged into one. We tokenized and converted text to lowercase. The preprocessing process is applied on both text and user profile location fields within the sample dataset.

3.3 Text Classification Model

Given an acquired tweet, the CNN text classification/predictive model determines whether or not the tweet is related to an emergency situation; and determines what category of incidents the tweets belong to if it incident-related. We use convolutional neural network (CNN), based on Kim's proposed architecture (Kim, 2014), for the classification tasks. By preparing incident-related and non-incident-related tweet samples as a training set, we produce a model to classify tweets automatically into incident-related and non-incident-related categories. The incident-related dataset is further trained to produce a model that classifies tweets automatically into the different categories of target emergency events. Figure 2 presents the CNN model architecture used to learn the internal representation of the text.

CNN system learns the key features and captures the most salient n-gram information by means of its convolution and max-pooling operations at different levels of abstraction automatically. Each word in the vocabulary V is represented by a D dimensional vector in a shared look-up table $L \in \mathbb{R}^{|V| x D}$where L is considered a model parameter to be learned (Nguyen et al., 2016). We initialize L randomly or using GloVe (Pennington et al., 2014) pretrained word embedding vectors

The input to our CNN is a matrix where each row is a real-valued vector representation of each word in the caption or tweet. Our word-vectors are obtained from a pretrained GloVe word embeddings. Given an input tweet $s = (w_1, \dots, w_T)$, we first transform it into a feature sequence by mapping each word token $w_t \in s$ to an index in L The look-up layer then creates an input vector$x_t \in \mathbb{R}^D$ for each token w_t which are passed through a sequence of convolution and pooling operations to learn high-level feature representations. This vector is then fed further down in the network to capture the most relevant features of the tweet and finally to the fully connected layer to perform prediction.

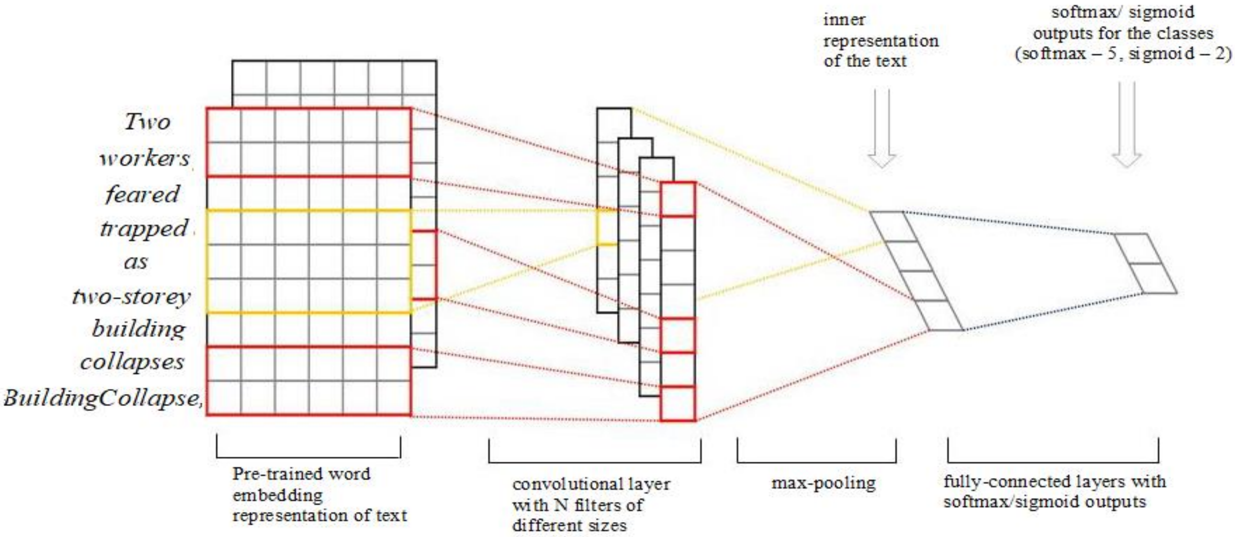

Figure 2: CNN Model Architecture for tweet classification (adapted from (Kim, 2014))

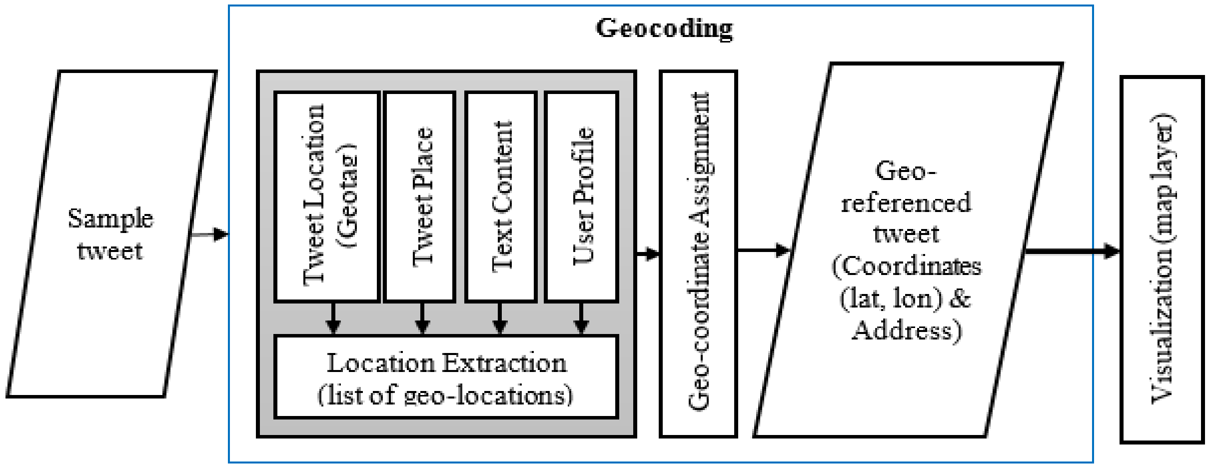

Figure 3: Model for the geographical location Inference.

3.4. Geocoding/Location Inference

Basically there are three metadata sources for geo-referencing tweets: Tweets that are geotagged with an exact location or Twitter Place using Geographical Positional System (GPS) information, mentioned location in tweets, and user profile location. The geocoding/location inference component deals with the extraction of predefined location references from each of these possible sources. Figure 3 shows the model of the geographical location estimator proposed in the system. When choosing a location for mapping, the system gives the highest priority in this order: place where the tweet is posted from (geographical coordinates of exact tweet location), the tweet Place, then the location mentioned in the tweet text, and then finally location relating to the user profile.

Algorithm 1 Emergency Event Detection and Location Estimation Algorithm

---

Input: Crawled from;
T: Tweet (contents and metadata)
Q: Query containing *K* keywords
Output: Location, *L*, of tweet [*location coordinates and address*]

---

1. Put a query Q containing K keywords, for some target emergency events, using Twitter Search API and obtain tweets, T.
2. For each tweet $t \in T$, filter it to extract tweet text, X, and perform preprocessing to normalize it for deep learning
3. Apply classification on X_t to obtain value $v_t = \{0,1\}$ where 1 is for incident-related event and 0 for non-incident-related event.
4. If $v_t = 1$ THEN perform step 5 through step10.
5. Apply classification on X_t to obtain incident type and its occurrence probability (confidence) using v_t, $t \in T$.
6. Set *locationFound* = *False*.
7. For each tweet $t \in T$ extract location information l_{info} from c, g, p, u where they exist in t [c = tweet text content, g = coordinates from GPS tweet location, p = coordinates of Place location and label, and u = User Profile Location].

$$l_{info} = \begin{cases} x, & if\ \exists\ x \in g_t, x \in p_t, x \in c_t, x \in u_t \\ & null,\ otherwise \end{cases} \quad (1)$$

8. Calculate the estimated location of the emergency event from l_{info}, $t \in T$ based on the following priorities: [Highest to lowest] : $g_t \to p_t \to c_t \to u_t$.
9. IF $(l_{info} \neq null)$ THEN

    IF $(g_t \neq null)$ THEN

        Assign the GPS coordinates g_t from $l_{info.}$: to location l_t
        [$l_t = l_{info.}(g_t)$] // *where* g_t *= coordinates(latitude, longitude)*
        Perform reverse geocoding to determine the address of the location:
        [*location = reverseGeocode(*l_l*)*]
        geocodeSource = 'Tweet GPS Location [lat,lng]'
        Set *locationFound* = True

    ELSE IF $(p_t \neq null)$THEN

        Assign the bounding box coordinates p_t from $l_{info.}$: to location l_t
        [$l_t = l_{info}(p_t)$ // *where* p_t = *bounding_box_coord(lat, lon)*]
        Perform reverse geocoding to determine the address of the location:
        [*location = reverseGeocode(*l_t*)*]
        geocodeSource = 'Tweet Place'
        Set *locationFound* = True

    ELSE IF $(c_t \neq null)$ THEN

        Parse text and resolve presence of entities $c_l \in c_t$ such as name of towns, communities, institutions, places, events, etc. that occur in the message by filtering $T_t \in \mathbb{R}^D$.
        IF found THEN

            Assign location in text, $c_l \in c_t$ to location $\boldsymbol{l_t}$
            Perform forward geocoding to assign coordinates to location, l_t,
            [*location = geocode(*$\boldsymbol{l_t}$*)*]
            geocodeSource = "Tweet Text"
            Set *locationFound* = True

        ENDIF

    ELSE // $(u_t \neq null)$

        Parse and resolve presence of address, u_l in the user profile, u_t.
        IF found *THEN*

            Assign location in text, $c_l \in c_t$ to location l_t

```
                        Perform forward geocoding to resolve the location,
                        [location = geocode(l_t)]
                        geocodeSource = ' Profile Location'
                        Set locationFound = True
                    ENDIF
                ENDIF
            ENDIF
10.     If locationFound THEN
                geoCoordinates = location.latitude, location.longitude
                geolocationAddress = location.address
                Visualize location on map
                (optionally) Send alert/notification e-mails to the appropriate agencies.
        ENDIF
```

4. EXPERIMENTS AND EVALUATION

4.1 Experimental Setup

In this section, we describe the experiment setup, results and evaluation of real-time tweet classification and location estimation. In the real time processing, the tweets are classified and geocoded as soon as they are crawled (acquired). We follow the framework shown in Figure 1 to learn the information contained in historical data and apply the algorithms in real-time**.** Experiments were run on a machine equipped with an Intel(R) Core(TM) i6-2520M and 2.50GHz CPU with 2 cores with 16GB RAM running a Microsoft Windows 10 64-bit operating system. All the models are built and implemented using Python 3.6 based on Keras deep learning library for Python running on tensor flow framework.

The algorithm for the real-time Emergency Event Detection and Location Estimation is shown in Algorithm 1. We first search for tweets *T* including the query *Q with K set of* keywords from Twitter using Twitter Streaming API for the target emergency events. We created a decoupled twitter stream using Kafka to collect the tweets. After classification and obtaining an incident related event within the target incident group, the location information of each tweet is used for location estimation of the target event.

Table 4: Dataset Statistics

Class	Dataset Size: Layer I Classification	Dataset Size: Layer II Classification
Class 1	190986	38266
Class 2	190576	38804
Class 3	-	37815
Class 4	-	37900
Class 5	-	38201
Total	383562	190986

4.2. Evaluation of Tweet Classification

For classification of tweets, we prepared 383562 data samples as a training set for the first Layer classifier and 190986 incident-related samples as the training set for the second Layer classification. The breakdown of the number of the samples for each class for each of the classification processes is shown in Table 4. For Layer I, classification has two classes (class1 –

incident and class 2 – non-incident). Layer II classification has five classes (1 – accident, 2 – crimes and civil disorder, 3 – damaged infrastructure, 4 – fire, and 5 – flood). We used 80% of the dataset as the training set and 20% as the test set. The validation set is 20% of the training set. In all experiments, the models (and hyper parameters) are tuned on the validation (development) set and the final performance evaluated against the test set. We trained the models by optimizing the cross entropy using the Adam optimizer and a maximum of 50 epochs. We used the pre-trained 300-dimensional GloVe (global vector for word representation) word embeddings/vectors that were pretrained on two billion tweets and 400,000-word vocabulary (Brownlee, 2017) to initialize the weight of the embedding layer.

We experimented with baseline and different CNN architectures as follows:

i) Baseline model: Support Vector Machine (SVM) trained and tested to validate the accuracy advantage of our models using bag-of-words feature extraction.
ii) CNN_I: A CNN model where all words are randomly initialized and then modified during training.
iii) CNN_{II}: A CNN model with pre-trained vectors from Glove for word embedding. All word-including the unknown ones that are randomly initialized – are kept static and only the other parameters of the model are learned.
iv) CNN_{III}: Same as in CNN_{II} but with parameters of the model being learned.

The precision, recall, and F_1-score are calculated as the measurements of performance.

Tables 5 and 6 show the performance evaluation results for layers 1 and 2 classifications respectively.

Table 5: Layer I Performance Evaluation (%)

Model	**Precision**	**Recall**	**F_1**
Baseline (SVM)	72.03	73.34	72.65
CNN_I	76.04	77.40	76.62
CNN_{II}	80.50	80.23	80.36
CNN_{III}	83.57	86.78	85.11

Table 6: Layer II Performance Evaluation (%)

Model	Precision	Recall	F_1
Baseline (SVM)	71.55	69.80	70.65
CNN_I	83.02	86.21	84.58
CNN_{II}	85.17	86.75	85.94
CNN_{III}	86.70	86.48	86.57

As shown in the tables, the best CNN model performs better than the baseline model by a significant margin of 12.46% and 15.92% for layer 1 and layer 2 respectively. For the last two models, CNN_{II} and CNN_{III} there were some gains over $CNN_{I,}$ obviously due to the use of the pretrained vectors. There was however, no significant difference between CNN_{II} where words were kept static and CNN_{III} where pretrained vectors were fine tuned for each task in layer II as against a 4.75% difference in layer I classification. The final F_1-score of the whole classification framework is 86.57%, a little above the closest performing model, CNN_{II}, with 85.94% F_1 score. The pretrained input word embeddings enjoy great success in the experiments, even without fine-tuning. This proves that the pre-trained word vectors have good generalization and adaptability in different classification tasks. Again, the deep neural network models perform better than non-neural models for both classification scenarios.

4.3 Evaluation of Spatial Estimation

Figure 4 shows tweets streamed and collected for analysis by the time of the day. A summary of the data acquisition and geocoding results is shown in Table 7. Among the 18542 tweets, about 20% of our acquired tweets are relevant to incidents. Only 558 tweets can imply meaningful incidents with accurate time and location and can be geocoded, representing about 15% of the incident related tweets. 59 tweets contain both text and images. None of the tweets contains image only.

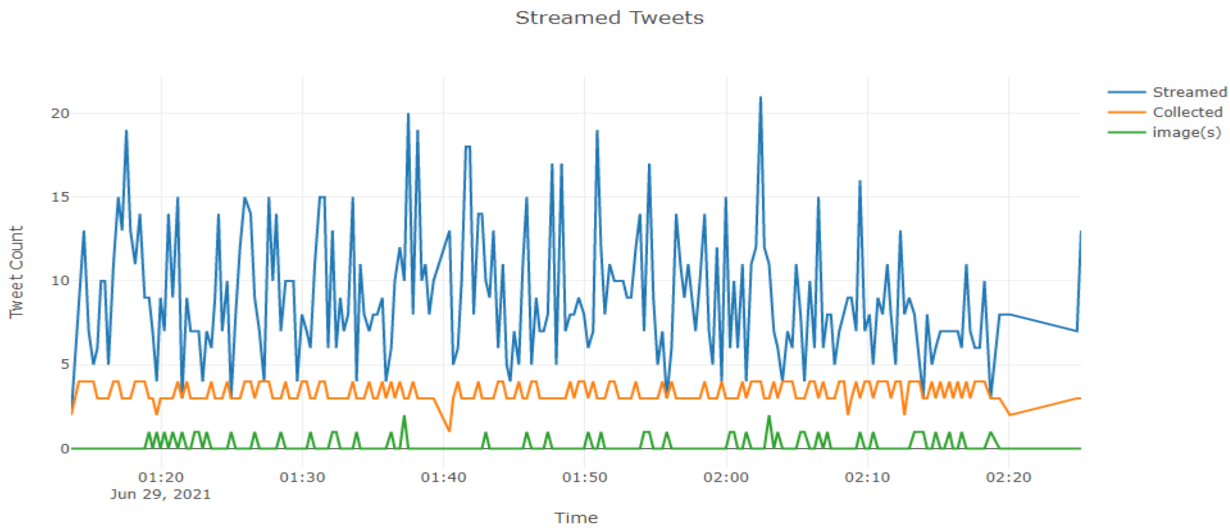

Figure 4: Tweets by the time of the day

Table 7: Results for the real-time experiment

Category of tweets	statistics
All tweets acquired	18542
Incident related tweets	3776
Geo-incident related tweets	558
Tweet with text and images	59
Tweet containing only image (without text)	0

Table 8 shows results of the application of the method on the sample tweets. The “Tweet ID” field is the unique identifier of a tweet assigned by Twitter. The “Geocode source” field indicates the location-related element, which is determined as the suitable element for location inference by the method. The “Inferred Location” field and its subfields (“Latitude” and “Longitude”) show the coordinates of the inferred location. The “Incident type” and “Confidence” fields are the detected and classified incidents and their confidence (probability of occurrence) respectively.

5. NOTIFICATIONAND REPORTING SYSTEM

We developed an emergency situation alert and reporting system using our methodology of emergency event detection and location estimation. A system screenshot is depicted in Figure 5. Users can see the detection of past and real-time target emergency events. The location estimation of each message collected and analyzed is used to produce a map layer to visualize where messages were published or are referring to. This can be seen in Figure 6 where OpenStreetMap is used as the base map. The system registers e-mail addresses and other details of the agencies that respond to the events to receive alerts/notices of occurrence of detected events for timely response.

Table 8: Results of the application of the method on sample tweets.

		Inferred Location				
Tweet ID	Tweet time	Latitude	Longitude	Geocode Source	Incident Type	Confidence
1432690427799429125	8/31/2021 14:03	12.75035	122.7312	Tweet Text	fire	97.05
1432690428461998085	8/31/2021 14:03	12.75035	122.7312	Tweet Text	crime_n_civil_disorder	76.78
1432690429661618177	8/31/2021 14:03	39.78373	-100.446	Profile Location	crime_n_civil_disorder	56.10
1432690433197441024	8/31/2021 14:03	36.15865	128.2573	Tweet Text	fire	99.47
1432690439497281538	8/31/2021 14:03	39.78373	-100.446	Profile Location	crime_n_civil_disorder	81.44
1432690448322154496	8/31/2021 14:03	52.505	-1.9644	Profile Location	accident	74.28
1432692174609604608	8/31/2021 14:10	51.4538	-2.5973	Profile Location	accident	99.80
1432692185648943110	8/31/2021 14:10	40.64606	-111.498	Profile Location	fire	99.65
1432692196638171136	8/31/2021 14:10	35.77301	-86.282	Profile Location	accident	99.46
1432692196935798786	8/31/2021 14:10	34.85872	136.8114	Tweet Text	accident	99.89
1432692207362916358	8/31/2021 14:10	32.44645	-99.7476	Profile Location	fire	95.34
1432693762510049280	8/31/2021 14:16	38.89499	-77.0366	Profile Location	fire	99.97
1432693762715570183	8/31/2021 14:16	39.78373	-100.446	Profile Location	crime_n_civil_disorder	93.50
1432693780189224960	8/31/2021 14:16	17.17505	95.99997	Profile Location	crime_n_civil_disorder	51.96
1432693787919192065	8/31/2021 14:16	40.71273	-74.006	Profile Location	crime_n_civil_disorder	94.41
1432693799780708353	8/31/2021 14:16	19.754	96.1345	Profile Location	crime_n_civil_disorder	93.97
1432695573195460613	8/31/2021 14:23	43.0692	10.87667	Profile Location	crime_n_civil_disorder	65.65
1432695574562840578	8/31/2021 14:23	12.75035	122.7312	Tweet Text	crime_n_civil_disorder	76.78
1432695590262034440	8/31/2021 14:23	27.94776	-82.4584	Profile Location	crime_n_civil_disorder	80.87
1432695610956734464	8/31/2021 14:23	30.33084	71.2475	Profile Location	crime_n_civil_disorder	69.22
1432695615008481282	8/31/2021 14:23	39.78373	-100.446	Profile Location	crime_n_civil_disorder	80.59
1432697358052102145	8/31/2021 14:30	35.45968	-97.6188	Tweet Coordinates	accident	92.34
1432697358115037190	8/31/2021 14:30	29.57435	-98.3256	Tweet Coordinates	accident	96.23
1432697358245113864	8/31/2021 14:30	35.12175	-80.9088	Tweet Coordinates	accident	91.34
1432697358349914121	8/31/2021 14:30	33.46282	-112.214	Tweet Coordinates	accident	94.09
1432697358358392834	8/31/2021 14:30	36.06691	-86.6868	Tweet Coordinates	accident	99.78

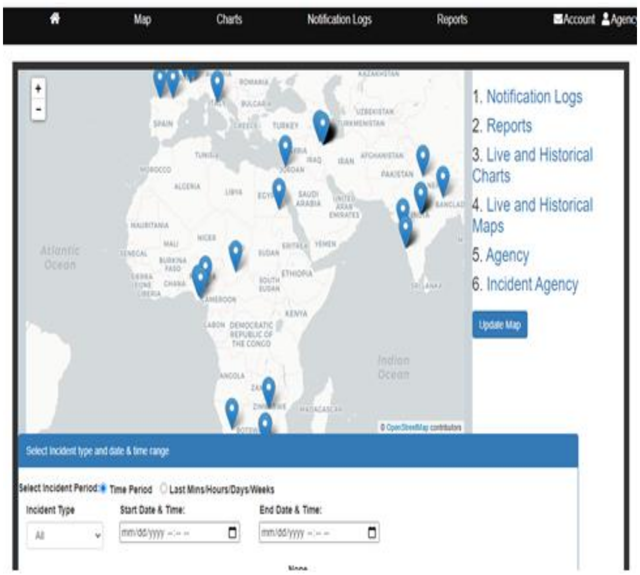

Figure 5: Screenshot of the emergency situation alert and reporting system

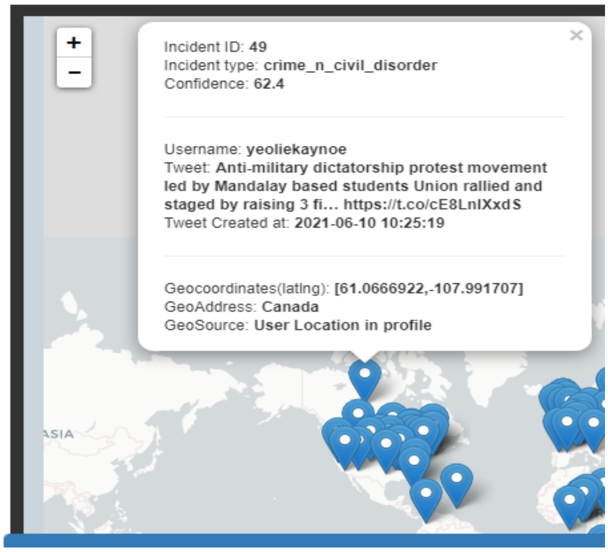

Figure 6: Pop-up of Incident geolocated and visualized on the map

6. CONCLUSION

We investigated the real-time nature of Twitter, a Microblogging site, with particular attention to emergency events detection. We presented an example that leverages the real-time nature of Twitter to make it useful in solving an important social problem: emergency/crisis situations. Tweets were collected in real-time using Apache Kafka technology to allow for the collection and processing of the raw data (tweets) from twitter in real-time without worrying about the stream getting disconnected due to overload. Tweets were classified in the first layer to determine whether or not the tweet is incident-related. The second layer determines the incident type of the tweet, if it is incident-related.

We estimated the geographical locations of events that are detected in tweets. We presented an approach that select one place as the geographic location of the target event which the tweet described based on the combination of the different kinds of locations we have got from geo-tags, user profiles, and the content from tweet text.

As an application, we developed an emergency alert and reporting system, which notifies the relevant agencies promptly of an emergency event.

REFERENCES

Ajao, O., Hong, J., and Liu, W. (2015). A survey of location inference techniques on twitter. *Journal of Information Science*, *41*(6), 855–864.

Ao, J., Zhang, P, and Cao, Y. (2014). Estimating the Locations of Emergency Events from Twitter Streams. *2nd International Conference on Information Technology and Quantitative Management*, 31, 731-739. doi: 10.1016/j.procs.2014.05.321

Bo, H., Cook, P., and Baldwin, T. (2012). Geolocation prediction in social media data by finding location indicative words. *Proceedings of COLING 2012: Technical papers*, 1045–1062.

Brownlee, J. (2017). A Gentle Introduction to Transfer Learning for Deep Learning. https://medium.com/analytics-vidhya/introduction-to-word- embeddings-c2ba135dce2f.

Cheng, Z., Caverlee, J., and Lee, K. (2010). You are where you tweet: a content based approach to geo-locating twitter users. *Proceedings of the 19th ACM international conference on information and knowledge management*, 759–768.

Ciulla, F., Mocanu, D., Baronchelli, A., Gonçalves, B., Perra, N., and Vespignani, A. (2012). Beating the news using social media: The case study of american idol. *EPJ Data Science, 1*(1), 1.

Dou, W., Wang, X., Skau, D., Ribarsky, W. and Zhou, M.X. (2012). Leadline: Interactive visual analysis of text data through event identification and exploration. *Proceedings of the IEEE Conference on Visual Analytics Science and Technology* (VAST).

Eskesen, S. (2017). Improving product categorization by combining image and title. Master Thesis.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). Deep Learning. MIT Press online. http://www.deeplearningbook.org.

Kelly, S., Zhang, X. and Ahmad, K. (2017). Mining Multimodal Information on Social Media for Increased Situational Awareness. *Proceedings of the 14th ISCRAM Conference* – Albi, 613-622.

Laylavi, F., Rajabifard, A. and Kalantari, M. (2016). A Multi-Element Approach to Location Inference of Twitter: A Case for Emergency Response. *International Journal of Geo-Information*

Lingad, J., Karimi, S., Yin, J. (2013). Location extraction from disaster-related microblogs. *Proceedings of the 22nd international conference on World Wide Web companion*. 1017–1020.

Mahmud, J., Nichols, J., and Drews, C. (2014). Home location identification of twitter users. *ACM Transactions on Intelligent Systems and Technology*, **5**, 47.

Morstatter, F., Pfeffer, J., Liu, H., and Carley, K. M. (2013). Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. *arXiv preprint* arXiv:1306.5204

Nguyen, D.T., Al-Mannai, K. A. A, Joty, S., Sajjad, H., Imran, M. and Mitra, P. (2016). Rapid Classification of Crisis-Related Data on Social Networks using Convolutional Neural Networks. arXiv preprint arXiv:1608.03902

Oku, K., Ueno, K., and Hattori, F. (2014). Mapping geotagged tweets to tourist spots for recommender systems. *Proceedings of the IIAI 3rd international conference on advanced applied informatics (IIAIAAI),* 789–794.

Paul, M. J., and Dredze, M. (2011). You are what you tweet: Analyzing twitter for public health. *ICWSM, 20,* 265–272.

Pennington, J., Socher, R. and Manning, C.D. (2014). Glove: global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.*

Ritter, A., Etzioni, O. and Clark, S. (2012). Open domain event extraction from twitter. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining.*

Robinson, B., Power, R. and Cameron, M. (2013). A sensitive twitter earthquake detector. *Proceedings of the 22nd International Conference on World Wide Web.*

Sakaki, T., Okazaki, M. and Matsuo, Y. (2010). Earthquake Shakes Twitter Users: Real-Time Event Detection by Social Sensors. *Proceedings of the 19th International Conference on World Wide Web*, 851-860.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2013). Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Trans. Knowl. Data Eng.* 25, 919–931. doi:10.1109/TKDE.2012.29

Schulz, A., Ristoski, P., and Paulheim, H. (2013). I see a car crash: real-time detection of small scale incidents in microblogs, *The Semantic Web: ESWC 2013 Satellite Events*, Montpellier, France, 7955, 2–33.

Schulz, A., Mencia, E. L., and Schmidt, B. (2016). A Rapid-Prototyping Framework for Extracting Small-Scale Incident-Related Information in Microblogs: Application of Multi-Label Classification on Tweets. *Information Systems*, 57, 88–110. DOI: 10.1016/j.is.2015.10.010.

Singh, J.P., Dwivedi, Y.K., Rana, N.P. Kumar, A. and Kapoor, K.K. (2019). Event classification and location prediction from tweets during disasters. *Ann Oper Res* 283**,** 737–757. https://doi.org/10.1007/s10479-017-2522-3

Skoric, M., Poor, N., Achananuparp, P., Lim, E.-P., and Jiang, J. (2012). Tweets and votes: A study of the 2011 Singapore general election. *Proceedings of the 45th Hawaii international conference on system science (HICSS)*, 2583–2591.

Wang, D., Mao, K., and Ng, G-W (**2017**) Convolutional Neural Networks and Multimodal Fusion for Text Aided Image Classification. *Proceedings of the 20th International Conference on Information Fusion*, Xi'an, China**.**

Yin, J., Lampert, A., Cameron, M., Robinson, B., and Power, R. (2012). Using social media to enhance emergency situation awareness. *IEEE Intelligent Systems*, 27(6), 52–59.

Zheng, X, Han, J. and Sun, A. (2018) A survey of location prediction on twitter, *IEEE Transactions* on *Knowledge* and *Data Engineering*, 30 (9), 1652-1671. doi:10.1109/TKDE.2018.2807840