FRIDZ: A Framework for Real-time Identification of Disaster Zones

Abhisek Chowdhury*

Rajendra Mishra School of Engineering Entrepreneurship
IIT Kharagpur, India
abhisekchowdhury@iitkgp.ac.in

Abstract. Social media feeds are rapidly emerging as a novel avenue for the contribution and dissemination of geographic information. Among which Twitter, a popular micro-blogging service, has recently gained tremendous attention for its real-time nature. For instance, during floods, people usually tweet which enable detection of flood events by observing the twitter feeds promptly. In this paper, we propose a framework to investigate the real-time interplay between catastrophic event and peoples' reaction such as flood and tweets to identify disaster zones. We have demonstrated our approach using the tweets following a flood in the state of Bihar in India during year 2017 as a case study. We construct a classifier for semantic analysis of the tweets in order to classify them into flood and non-flood categories. Subsequently, we apply natural language processing methods to extract information on flood affected areas and use elevation maps to identify potential disaster zones.

Keywords: Flood, Twitter, Location estimation, Clustering, NLP

1 Introduction

Traditional methods for identification of catastrophic event zones are quite time-consuming and lacks accuracy regarding the spatial and temporal dynamics. However, emergency situations such as floods require quick responses from the national organizations and governing bodies. Peoples' reaction during flood such as tweets on Twitter provide an opportunity to identify the disaster zones in real-time.

After its launch on July 2006, Twitter users have proliferated. As of the third quarter of 2017, this micro-blogging service is estimated at 330 millions[1] monthly active users worldwide. Although there are other real-time micro-blogging services such as *Tumblr*[2], *Flickr*[3], we especially investigated Twitter because of its vogue and enormous data volume.

[*] This paper has been accepted for Oral presentation at PReMI 2019, Assam, India.

[1] http://tiny.cc/nw0sbz

[2] https://www.flickr.com/

[3] https://www.tumblr.com/

In this paper, we propose a framework to identify catastrophic event zones (eg. flood as a case study) in real-time using the social media platform, Twitter and auxiliary spatial data resources. Data from geographic information system (GIS) and twitter messages were compiled and evaluated using data mining techniques for the state of Bihar in India during the 2017-18 Monsoon season. Furthermore, auxiliary spatial data have been used for demarcation of the disaster zones. The outcome of this procedure is correlated with official flood reports[4] to visualize a comparison between them. The summary of major contributions of our work is presented as follows:

- Estimation of location information from semantic analysis of tweets.
- An approach to utilize extracted location information with auxiliary spatial data to identify disaster zones.

This paper is organized as follows: Section 2 contains the related works in this field. Section 3 elaborates the detailed methodology of the proposed framework. Case study and results have been presented in Section 4. Finally, Section 5 concludes our work and enlist the future scope.

2 Related Works

In recent years, social media have been an inevitable part of everyday life across many parts of the world. Various researches have been conducted on Twitter data. Mei et al. [1] concentrated on blogs and inspected their spaio-temporal patterns. Sakaki et al. [2] utilized tweets to produce a spatio-temporal semantic model for detection of earthquake epicentre and trajectory for a typhoon. Middleton et al. [3] computed a final n-gram token from a sequential combination of the 1-gram tokens from which locations were extracted by using Natural Language ToolKit's (NLTK's) Treebank word tokenizer. Brunsting et al.[4] showed an approach of geographical information tagging into textual documents. Salehi et al. [5] described how a name can reveal one's location. Chi et al. [6] explored the importance of location indicative words for location prediction in a tweet. Various works showed effectiveness of GIS and machine learning tools for surveillance of real-time events like influenza outbreaks [7] and traffic congestion detection with optimal path recommendation [8].

Numerous techniques have been explored for spatial clustering such as density based methods, partitioning methods, grid-based methods, model-based methods, hierarchical methods etc. For instance, SNN [9] was also established to cluster the earth science data. Moreover, DBSCAN [10] and IncrDBSCAN [11] have been devised to study the spatial data sets as well. Birant and Kut brought ST-DBSCAN [12] to determine clusters on spatio-temporal data. They also discovered noise objects when clusters of diverse densities exist. *Density factor* was also introduced which is the degree of the density of the cluster. Later, scientists Ankerst et al. [13] contemplated OPTICS for suitable parameters estimation in DBSCAN. J. Wang et al. invented DBCTAR [14] for analysis of traffic accident risk.

[4] https://reliefweb.int/report/india/situation-report-1-bihar-flood-2017

3 Framework for Disaster Zone Identification

This section illustrates the methodology of our framework by describing the techniques used to implement the idea which has been schematically rendered in Figure 1.

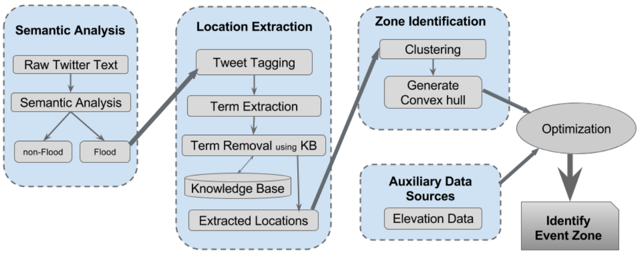

Fig. 1. Overview of the FRIDZ framework

3.1 Semantic Analysis

To uncover a target event, we searched the twitter streams with the keyword *'flood'*. By observing the collected data set, we found that many tweets containing the keyword *'flood'* actually do not indicate a flood event. For instance, someone might tweet – *"Today we were attending the International Flood Conference at Bangalore..."*

Therefore, it is essential to clarify that a tweet is truly indicating a real-world flood occurrence, which is denoted as a 'VALID' class. In this section, we have implemented SVM classifier to eliminate such misleading tweets that do not arrive in hinting a real-world cases of flood. Table 1 shows some example of tweets which contain the keyword *'flood'* and the decision of their validity to perform the classification task has also been defined. In many cases, these examples show that the tweets can be classified by recognizing one of the positive indicators like *'help', 'deaths', 'risk'* or one of the negative indicators like *'joke', 'lectures'*. The task of attaching class labels either *VALID* or *INVALID* to a tweet is modelled as a binary classification task. The features which are pertinent to this task are following:

- **Feature 1:** Number of words in a tweet.
- **Feature 2:** Position of the keyword *flood* in a tweet.
- **Feature 3:** Words (eg. 'help', 'deaths', 'risk') presented in a tweet.
- **Feature 4:** n-grams (eg. 'affected parts', 'trapped people') in a tweet.

In order to assign numeric values to n-grams eg. unigram, bi-gram, tri-gram features, we have utilized the "TF-IDF" (Term Frequency, Inverse Document Frequency). It is a way to measure the importance of *terms* in a message depending on how periodically they arrive across multiple messages. Thus, frequent words like 'the' and 'for' will be scaled down due to the existence in many messages and words that come out frequently in a single message will be scaled up.

No.	Tweets	Class Labels
1.	Over 28,000 #flood victims have been rescued or evacuated from various affected parts of the country till now: #NDRF #BiharFloods.	**VALID**
2.	A man interrupted one of the Buddhas lectures with a flood of abuse. Buddha waited until he had finished and then asked him...	**INVALID**
3.	West champaran Highwater is at risk of flood @ndmaindia	**VALID**
4.	How the instant joke creators earn money? Like flood of msgs aftr India Pakistan match.	**INVALID**

Table 1. Tweets with manually labelled class

3.2 Location Extraction

Availability of geo-tagged twitter data is very rare in the practical scenario. The geo-tagged tweets are considered directly for location (coordinate) extraction, and non-geo-tagged ones are further processed. Our approach makes use of NLP methods by using a state-of-the-art part-of-speech (POS) tagger along with *Named Entity Recognizer (NER)* to find blocks of text which may refer to potential locations. The entire text message is tagged by both these taggers. Based on the output, the algorithm creates the set of terms 'T' which represents the potential location references in the tweets.

Text Message Tagging: The POS tagger assigns tags to each word of the short twitter texts. Among various kinds of available POS tags, we focus on a subset of POS tags which are relevant to our problem and are represented in table 2. The NER tagger tags each block of text messages with one of the four possibilities: "Location", "Person", "Organization", "O" (for Other).

Sl. No.	POS Tag	POS Tag Description	Grouping
1	NN	Noun, Singular	Noun
2	NNS	Noun, Plural	Noun
3	NNP	Proper noun, Singular	Noun
4	NNPS	Proper noun, Plural	Noun

Table 2. Parts-of-Speech tags used in this approach

Term Extraction: After each block of text messages has been tagged by those taggers, we have our term set, 'T' which contains all potential location names of the target event. Initially, only noun tags from the POS tagger are considered in term set T. Thereafter, while reducing the set T in a subsequent step, words with "Location" tags from NER are retained. A word is considered as a potential location if the word is tagged by any POS tags from table 2 and a "Location" tag from the NER. Some locations may have multiple words in their name. We have considered these multi-word locations while building the set 'T'. For example, if the word sequence "New Delhi" is in the text message, with each of the two words tagged as a noun (note that the POS actually tags 'New' as a noun due to the capitalization), then three terms will be added to the set T : *'New'*, *'Delhi'* and *'New Delhi'*. The term set 'T' is not finalized yet. Some of the terms added in this section will be removed for better efficiency, as described in the next section.

Term Removal using Knowledge Base: In the resultant term set 'T', each item can be a place name or non place name since relying on POS and NER tags are not sufficient enough. To remove this anomaly, extracted locations are searched in a knowledge base (Google Map API [5]). Forward geocoding mechanism eliminates the non-place names from the term set 'T'. The amount of time required for the knowledge base searches grows linearly with the cardinality of term set 'T'. This phase converts the contents of set 'T' from place names to latitude-longitude pairs along with the frequency of mentioning of that location by the users. Now we get the new set of locations 'L' from 'T' and each element of set 'L' will be denoted by a tuple 'L_p' comprising of latitude, longitude and frequency of a particular location p.

$$L_p = \langle\ p_{lat}, p_{lon}, p_{freq}\ \rangle$$

3.3 Disaster Zone Identification

The goal in this portion is to find alluring clusters from the location data set 'L' obtained from the previous section. Since there are various alternatives for clustering methods, we opt for an algorithm which satisfies following criteria: (i) The selection for the number of clusters should be automated. (ii) Since our only desire is to identify event zones, the algorithm should eliminate points with minor density (frequency of mentioning) measures.

Moreover, such detection is not only a statistical demand but also for a real-time estimation. Due to which we propose a modified spatial clustering method *DBCTweeFr* (Density-Based Clustering for Tweet Frequency) which is a wrapper over the DBSCAN algorithm [10] (Density-Based Spatial Clustering of Applications with Noise) by considering the frequencies of mentioning for each location to group the potential data points from the distribution of the target events which is summarized in Algorithm 1. Data has been filtered out considering frequencies of locations using a threshold fr_{th} which has been empirically set. Furthermore, convex-hull have been generated for all detected clusters to identify the event zones.

3.4 Disaster Zone Optimization

The identified event zones in shape of polygons are further optimized by utilizing the elevation data sets, '$ELEV$'. The pseudo code for this optimization technique is summarized in Algorithm 2. Elevation which is basically an integer value, for each boundary points of the polygons is calculated from the '$ELEV$'. Subsequently, the Moore neighbourhood [6] set of radius 1 is formed which consists of the 8 neighbourhood data points for each boundary point of the polygons.

[5] https://developers.google.com/maps/documentation/javascript/examples/geocoding-simple

[6] https://en.wikipedia.org/wiki/Moore_neighborhood

Algorithm 1 DBCTweeFr, Density Based Clustering for Tweet Frequency

```
Input: $L = \{L_1, L_2, ..., L_p, ..., L_n\}$, Set of all potential locations.
$eps$: Radius for the neighborhood of point.
$min\_pts$: Minimum number of points in the given neighborhood.
$fr_{th}$: Tweet frequency threshold for each locations which is empirically set.
Output: $\rho$, Set of polygons for all clusters.
Procedure:
$\nu \leftarrow NULL$ , $\rho \leftarrow NULL$
for all location $l \in L$ do
    if $l.p_{freq} > fr_{th}$ then
        $\nu \leftarrow \nu \cup l$
    end if
end for
if $\nu \neq NULL$ then
    $C \leftarrow DBSCAN(eps, min\_pts, \nu)$
end if
for all cluster $\zeta \in C$ do
    $\psi \leftarrow conv\_hull(\zeta)$        //$conv\_hull()$ generates convex-hull over the cluster
    $\rho \leftarrow \rho \cup \psi$
end for
return $\rho$
```

The Moore neighborhood of a point P is the set of 8 points which share a vertex or edge with that point.

The elevation for each element of the neighbourhood set is compared with the elevation of the actual boundary point and if it is found that the elevation of any one member of the neighbour set is lesser than that of the boundary point then we update the position of the boundary point to that particular low elevation point. Otherwise, we will keep the actual data point untouched. This monotonous process is continued for each boundary point of all polygons.

4 Case Study

In the time span from 24th July 2017 to 13th September 2017, extreme raining affected large part of eastern India which includes Bihar state. Out of 38 districts of Bihar, 14 districts were affected. Kishanganj, Araria, Katihar and Purnia were the worst affected areas. We have chosen Bihar, India (25.0961° N, 85.3131° E) as our study area.

4.1 Data Sets

Twitter data set For this study, we collected *tweets* using the python based Twitter API, *tweepy* [7], which provides access to a 1% sample of the real-time twitter streaming by querying with the keyword *'#flood'* during the interval from 24 July 2017 to 13 September 2017. The collected data set consists of 1,00,790 *tweets* within the province of India.

[7] http://www.tweepy.org/

Algorithm 2 Optimizing the event zone using elevation

```
Input: $\rho$, Set of polygons containing all the clusters.
$elev(pts)$ : Finds elevation of a point, $pts$
$min\_neighbor\_elev(pts)$ : Finds the minimum elevation from the Moore neighborhood set of point, $pts$ and also returns the coordinates of that point
$min\_elev$ : Stores the minimum elevation obtained from $min\_neighbor\_elev(pts)$
Output: $\rho_{opt}$, Optimized polygons for all clusters.
Procedure:
$\rho_{opt} \leftarrow \rho$
for all polygon $\varphi \in \rho_{opt}$ do
    for all point $pts \in \varphi$ do
        $min\_elev, \eta \leftarrow min\_neighbor\_elev(pts)$
        if $elev(pts) < min\_elev$ then
            $pts \leftarrow \eta$
        end if
    end for
end for
return $\rho_{opt}$
```

Auxiliary data sets We have used digital elevation model (DEM) of the Shuttle Radar Topography Mission (SRTM) [8] at 3 arc-second resolution as Elevation data sets to distinguish between the high altitude zone with the sea-level regions.

4.2 Experiments and Results

Semantic Analysis: The SVM classifier is trained by using 1,500 tweets which are randomly sampled from the data set. A group of 3 volunteers manually annotated the samples into both *VALID* and *INVALID* classes. Using 500 randomly sampled tweets in a test set, the classifier gives a precision score of *0.813* and a recall score of *0.677* which substantially signifies that the constructed model was adequate to detect most of the valid tweets precisely indicated by the decent recall score, but it occasionally incorrectly categories tweets as a *VALID* one indicated by the precision score.

Extracting Locations: At the end of the data collection period, only 8,063 geo-tagged tweets are available which constitute merely 8% of the total tweets. By further processing the non-geo-tagged tweets, the terms such as 'New', 'Delhi' and 'New Delhi' are identified as potential locations by the taggers. After which we searched the knowledge base with location name along with country code which is *'IN'* for *India*. This returns coordinates for *'Delhi, IN'* and *'New Delhi, IN'* but *NULL* for *'New, IN'* which subsequently helps us to remove the non-potential locations.

Disaster Zone Analysis: The algorithm *DBCTweeFr* eliminates the locations with lower mentioning frequency since some places are reported as target event location by very fewer number of users. Rest of the locations are clustered based on the location coordinates. The identified zones are highlighted by the *RED*

[8] https://lta.cr.usgs.gov/SRTM1Arc

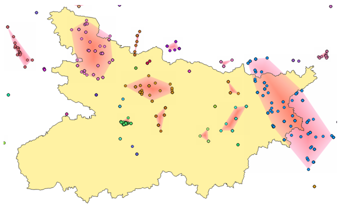

Fig. 2. Clustered locations

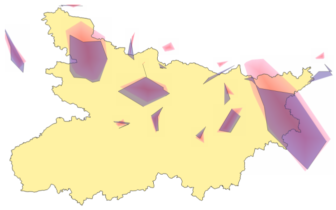

Fig. 3. Optimized event zones

polygons depicted in Figure 2. After which, the optimization algorithm 2 further shrink the identified zones using the concept – "the lesser the elevation of a point, the more prone to flood" and the outcome is represented in Figure 3 by the *VIOLET* polygons.

We have compared the identified disaster zones by our framework with officially reported areas which gives significant results with Jaccard Similarity Index [15] of 0.636 using Equation 1 represented in Figure 4.

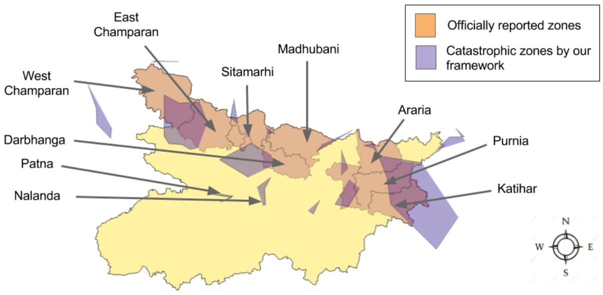

Fig. 4. Comparison with officially reported areas with the FRIDZ identified zones

According to the news report, Araria, Katihar and Purnia districts were the worst affected area by floods. Our framework successfully identified these areas as the disaster zones. Whereas, flood waters had also partially affected West Champaran, East Champaran, Madhubani, Patna, Nalanda, Sitamarhi and Darbhanga districts. In this case, our framework lacks efficiency in identifying some areas such as Darbhanga, Madhubani and Sitamarhi accurately. This highlights that our approach is applicable for identification of the flood zones.

$$JaccardSimilarityIndex(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

here A is set of Officially Reported Zones and B is set of 'FRIDZ' Identified Zones.

5 Conclusions and Future Works

In this work, we have presented a framework to identify disaster zones in real-time by integrating auxiliary spatial data with tweets. Our case study have shown that detection of flood events by constructing a classifier and geoparsing from twitter data are possible by exploiting the large open data resources. The approach has been successfully demonstrated for Bihar in India as a case study which gives a promising result.

In future, we would like to use unsupervised probabilistic models to measure flood vulnerability of a point instead of using elevation data directly. Also we have plan to work with heterogeneous data resources like meteorological data sets, road networks, census data for more accurate demarcation of disaster zones.

References

1. Mei, Q., Liu, C., Su, H., Zhai, C.: A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In: Proceedings of the 15th international conference on World Wide Web, ACM (2006) 533–542
2. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web, ACM (2010) 851–860
3. Middleton, S.E., Middleton, L., Modafferi, S.: Real-time crisis mapping of natural disasters using social media. IEEE Intelligent Systems **29**(2) (2014) 9–17
4. Brunsting, S., De Sterck, H., Dolman, R., van Sprundel, T.: Geotexttagger: High-precision location tagging of textual documents using a natural language processing approach. arXiv preprint arXiv:1601.05893 (2016)
5. Salehi, B., Hovy, D., Hovy, E., Søgaard, A.: Huntsville, hospitals, and hockey teams: Names can reveal your location. In: Proceedings of the 3rd Workshop on Noisy User-generated Text. (2017) 116–121
6. Chi, L., Lim, K.H., Alam, N., Butler, C.J.: Geolocation prediction in twitter using location indicative words and textual features. In: Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT). (2016) 227–234
7. Allen, C., Tsou, M.H., Aslam, A., Nagel, A., Gawron, J.M.: Applying gis and machine learning methods to twitter data for multiscale surveillance of influenza. PloS one **11**(7) (2016) e0157734
8. Ghosh, S., Chowdhury, A., Ghosh, S.K.: A machine learning approach to find the optimal routes through analysis of gps traces of mobile city traffic. In: Recent Findings in Intelligent Computing Techniques. Springer (2018) 59–67
9. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proc. 20th int. conf. very large data bases, VLDB. Volume 1215. (1994) 487–499
10. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. Volume 96. (1996) 226–231
11. Ertöz, L., Steinbach, M., Kumar, V.: Finding topics in collections of documents: A shared nearest neighbor approach. In: Clustering and information retrieval. Springer (2004) 83–103
12. Birant, D., Kut, A.: St-dbscan: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering **60**(1) (2007) 208–221
13. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: ordering points to identify the clustering structure. In: ACM Sigmod record. Volume 28., ACM (1999) 49–60
14. Wang, J., Wang, X.: An ontology-based traffic accident risk mapping framework. Advances in Spatial and Temporal Databases (2011) 21–38
15. Tan, P.N.: Introduction to data mining. Pearson Education India (2018)