The detection and location estimation of disasters using Twitter and the identification of Non-Governmental Organisations using crowdsourcing

Citation for published version:
Loynes, C, Ouenniche, J & De Smedt, J 2020, 'The detection and location estimation of disasters using Twitter and the identification of Non-Governmental Organisations using crowdsourcing', *Annals of Operations Research*, vol. N/A, pp. 1-33. https://doi.org/10.1007/s10479-020-03684-8

Digital Object Identifier (DOI):
10.1007/s10479-020-03684-8

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Annals of Operations Research

Noname manuscript No.
(will be inserted by the editor)

The Detection and Location Estimation of Disasters Using Twitter and the Identification of Non-Governmental Organisations Using Crowdsourcing

Christopher Loynes · **Jamal Ouenniche** ·
Johannes De Smedt

Received: date / Accepted: date

Abstract This paper provides the humanitarian community with an automated tool that can detect a disaster using tweets posted on Twitter, alongside a portal to identify local and regional Non-Governmental Organisations (NGOs) that are best-positioned to provide support to people adversely affected by a disaster. The proposed disaster detection tool uses a linear Support Vector Classifier (SVC) to detect man-made and natural disasters, and a Density-based spatial clustering of applications with noise (DBSCAN) algorithm to accurately estimate a disaster's geographic location. This paper provides two original contributions. The first is combining the automated disaster detection tool with the prototype portal for NGO identification. This unique combination could help reduce the time taken to raise awareness of the disaster detected, improve the coordination of aid, increase the amount of aid delivered as a percentage of initial donations and improve aid effectiveness. The second contribution is a general framework that categorises the different approaches that can be adopted for disaster detection. Furthermore, this paper uses responses obtained from an on-the-ground survey with NGOs in the disaster-hit region of Uttar Pradesh, India, to provide actionable insights into how the portal can be developed further.

Keywords Disaster Detection · Disaster Management · Location Estimation

Christopher Loynes
Edinburgh, Scotland, United Kingdom

E-mail: christopher.loynes@gmail.com

Jamal Ouenniche & Johannes De Smedt
University of Edinburgh Business School Management Science and Business Economics Group, 29 Buccleuch Place, EH8 9JS, Edinburgh, UK
E-mail: Jamal.Ouenniche@ed.ac.uk & Johannes.DeSmedt@ed.ac.uk

1 Introduction

This paper seeks to provide the humanitarian computing community with a supervised automated tool that is able to detect unseen disasters on Twitter, referred to as *disaster detection*. The paper attempts to address the research problem of detecting a disaster and accurately estimating its geographic location using information contained in tweets, whilst dealing with the large volume and velocity of content generated on Twitter.

A three-stage hybrid *disaster detection* tool is proposed, which uses a corpus of labelled tweets provided by [27] that covers 11 disasters across eight different disaster-types. Stage one involves normalising tweets. Stage two involves text classification. To determine the most suitable algorithm to use for classification, four multinomial text classifiers are assessed (referred to as 'testing'), namely a *Logistic Regression*, *Naive Bayes*, *Random Forest* and a linear *Support Vector Classifier (SVC)*. The top performing classifier and an appropriate benchmark are then selected for further analysis (referred to as 'evaluation'), which entails fitting the two classifiers on increasingly larger randomly sampled sub-sets of the corpus, to simulate the effect of streaming tweets from Twitter. Stage three involves clustering the classified tweets using Density-based spatial clustering of applications with noise (DBSCAN). This determines if a disaster has occurred and if so, estimates its geographic location.

To complement the proposed *disaster detection* tool, a prototype portal[1] has been built, for the purpose of harnessing the power of crowdsourcing to identify local and regional Non-Governmental Organisations (NGOs), that can help people adversely affected by a disaster. This prototype serves as a proof-of-concept, demonstrating how concerned individuals can communicate ideas to identify credible NGOs. The structure of the portal and how suggestions are ranked to identify those of high value are detailed in section 4. The motivation behind developing the portal is due to aid and disaster relief operations being largely uncoordinated in the immediate hours following a disaster. After a disaster has hit a region, most lives are saved by assistance provided from local communities and NGOs, and not by funds deployed by international aid agencies [29]. As such, if the proposed portal is effective at identifying local and regional NGOs in the immediate hours after a disaster has taken place, it may enable more effective disaster relief to be conducted.

The originality in proposing an automated *disaster detection* tool and the prototype portal is their potential synergy. The effective detection of a disaster, an accurate estimation of its location and the subsequent identification of local and regional NGOs, means NGOs best-positioned to help people affected by a disaster can begin administering disaster relief in a short period of time. The simple but effective relationship between the proposed *disaster detection* tool and the prototype portal can be seen in figure 1 below:

[1] https://s1719242.wixsite.com/website

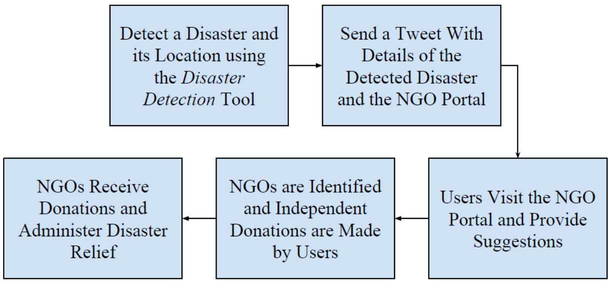

Detect a Disaster and its Location using the *Disaster Detection* Tool
Send a Tweet With Details of the Detected Disaster and the NGO Portal
Users Visit the NGO Portal and Provide Suggestions
NGOs are Identified and Independent Donations are Made by Users
NGOs Receive Donations and Administer Disaster Relief

Fig. 1: How the proposed *disaster detection* tool and NGO portal are inter-linked

In an effort to realise the potential of the prototype portal, an on-the-ground survey was conducted by visiting eight NGOs in Uttar Pradesh, India in June 2018. This is due to India being the third most affected country in the world by disasters in the last decade (measured by economic impact and the number of fatalities) and Uttar Pradesh is the most disaster-hit region in India [21]. The first and second most affected countries are Haiti and China, respectively. Since Haiti was largely affected by a single disaster in 2010 and the disasters in China are spread over too large a geographic area to visit, India was selected.

Also proposed in this paper is a generic *disaster detection* framework, which interlinks *disaster detection* approaches proposed by [5]:

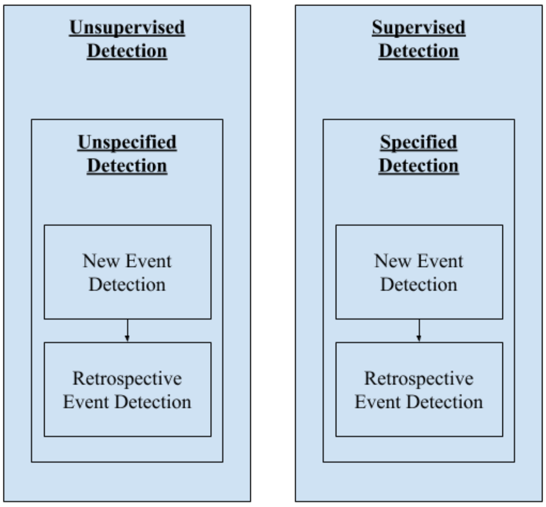


Fig. 2: Proposed generalised framework of *Disaster Detection* approaches

Figure 2 shows that *disaster detection* can be categorised into *Unsupervised Detection* and *Supervised Detection*. This is based on whether a labelled training dataset is used, which is the case for *Supervised Detection*. For *Unsupervised Detection*, *Unspecified Detection* is performed. This involves detecting patterns and features within tweets, since no specific information relating to a disaster is specified. Within *Unspecified Detection*, a new disaster or retrospective disaster can be detected, using *New Event Detection* and *Retrospective Event Detection*, respectively. The former involves detecting an unseen disaster. The latter involves identifying tweets that relate to a disaster that is known to have occurred already. This is often performed to obtain information that is of value to emergency responders. In the case of *Supervised Detection*, *Specified Detection* can be performed. This involves detecting a disaster using specific information, such as key words and locations. Within this, both *New Event Detection* and *Retrospective Event Detection* can be performed, for the same reasons just detailed. The aim of the proposed framework is to provide an overview of the different approaches that can be employed to perform *disaster detection*, which would facilitate the development of *disaster detection* tools in the future.

The remainder of this paper is structured as follows. Section 2 covers the literature pertaining to disaster detection based on various categorisations. Section 3 details the methodology implemented for the proposed *disaster detection* tool. Section 4 describes the structure of the prototype NGO portal. Section 5 outlines the empirical investigation and findings for the *disaster detection* tool, which covers ‘testing’ and ‘evaluation’, alongside the performance of the *DBSCAN* algorithm, used to detect a disaster and estimate its geographic location. This section also provides an overview of the prototype portal and areas for its development in the future. Lastly, section 6 provides a conclusion, which summarises the contributions made in this paper, alongside potential areas for further work.

2 Literature Review

In this paper, *disaster detection* is broken down into three sub-sections: *monitoring*, *location estimation* and *situational awareness*.

2.1 Monitoring

The act of *monitoring* involves continuously analysing tweets, with the aim of detecting a disaster, often in real-time. In the case of Twitter, the high velocity and volume of tweets that need to be processed means that any tool proposed for *monitoring* needs to be computationally cheap. The literature relating to *monitoring* has been categorised by technique: *filtering*, *clustering*, *classification* and *hybrids*. Given the complexity in reliably and accurately *monitoring* tweets, the vast majority of research analysed proposes hybrid techniques. This is largely due to hybrid techniques leveraging the strengths of one algorithm to compensate for the weaknesses of another.

2.1.1 Filtering

When *monitoring* Twitter to detect a disaster, *filtering* techniques can be implemented [45]. Despite *filtering* requiring an initial set of rules/features to be specified, such as key terms, venue and time, it is possible to develop an algorithm that can evolve and expand beyond those initially stated. [55] developed a 'semi-automatic' process to build and update an *ontology*, where an *ontology* defines the relationships between categories in a subject area, such as a disaster. The approach by [55] begins with a *seed ontology*, which is initially stated by the user. This is then extended using a semantic network called *ConceptNet5*[2] and a 'semi-automatic' updating process. The proposed method reduces the amount of manual effort required to define an *ontology*, whilst remaining accurate and concise. The approach does not run in real-time, however, it is argued by [55] that macro-event disasters, such as an earthquake or the outbreak of Ebola, are 'non-time location sensitive events'. This type of event is said to happen over a long period of time and affect a large geographic area. For these reasons, [55] suggest that tweets that relate to such events will be posted around the world, meaning the location they are sent from offers no value. Consequently, there is less emphasis on detecting an event immediately after it has occurred and a manually-intensive statistical approach that can detect *bursty* key terms is not a major drawback. A strong counterargument to this is the significant focus that almost every other paper surveyed in this literature review places on leveraging location-related information as quickly as possible.

An example of a three-stage unsupervised approach to *monitoring* that can be adapted for *disaster detection* is proposed by [39], which identifies emerging trends, provides analytics and accurate descriptions of each topic in real-time. The tool uses an algorithm called *QueueBurst* to detect and group key words that appear in a small period of time.

2.1.2 Clustering

Clustering algorithms that are computationally cheap and require no prior knowledge, such as the number of clusters, are well-suited to processing tweets in near real-time [5]. Consequently, *partitioning clustering* techniques, such as *K-means* and *K-median*, are not suitable, since the number of clusters (K) must be known in advance [3]. An example of a clustering algorithm that has been implemented for *monitoring* is *hierarchical clustering*. In the case of [36], a graph-based approach is implemented, which splits topical words into clusters. Disasters are subsequently tracked using *bipartite graph matching*, which is then summarised for manual observation. This type of *hierarchical clustering* technique is considered to perform poorly on a large scale, since it requires a full similarity matrix to be computed, which contains the pairwise similarity between each group, making it computationally expensive and inappropriate for real-time *disaster detection* [7, 13]. An alternative clustering approach to *monitoring* is proposed by [59], with a three-stage technique called *Event*

[2] www.conceptnet.io

Detection with Clustering of Wavelet-based Signals (EDCoW), which builds signals for individual words by applying *wavelet analysis* on the frequency-based raw signals of words.

Another clustering technique that has been used for *monitoring* is *DBSCAN*. This can be considered the state-of-the-art clustering technique in *disaster detection*, since its use of index structures for density estimation makes it scalable to large datasets [30]. For a cluster to be formed and a disaster to be detected, the neighbourhood of a specified radius must contain a minimum number of tweets. This approach treats outliers as noise, makes no assumption on the number of clusters and is capable of identifying randomly-shaped clusters. An example of *DBSCAN* in the domain of *monitoring* is *Tweet SCAN* [10], which uses the content, time, location and user-to-group relationship of a tweet. After normalising tweets, which involves case-folding, removing numbers, removing special characters and stripping out white spaces, the textual content is modelled through a probabilistic model called *Hierarchical Dirichlet Process* and the *Jensen-Shannon Divergence* is calculated for neighbourhood detection. The approach is largely based on the work of [49], however, *Tweet SCAN* only uses geotagged tweets. This makes it less effective than the *Incremental DBSCAN* technique proposed by [32], which uses both geotagged and non-geotagged tweets. Despite this, the work of [10] is an important step into using spatial, temporal, textual and user features for the task of *monitoring*. A similar *DBSCAN* method to that of [10] is proposed by [33] called *BursT*. The premise is similar to [32], as it seeks to detect and group emerging topics from real-time tweets, however, it has an increased focus on the weighting of *bursty* terms.

2.1.3 Classification

Text classification is commonly used for *monitoring*, since the content generated on Twitter is predominantly text-based. The wealth of text classifiers available means that those that are computationally inexpensive are most suitable for *monitoring*. One approach that can help reduce the computational cost of detecting events in real-time is by specifying features. This is most commonly a known or planned social event, which can be specified to differing degrees of specificity, using content and/or metadata, such as venue, time and location [5]. In the case of [45], a *monitoring* tool is proposed that is specific to earthquakes. Each user who tweets about an earthquake is treated as a sensor. To process Japanese tweets, morphological analysis is conducted using *Mecab*[3]. English tweets are normalised by performing stop-word elimination and stemming. Using the normalised tweets, the tool proposed by [45] detected 96% of earthquakes that were greater than a Japan Meteorological Agency (JMA) seismic scale of three or more. The approach proposed by [45] was developed further by [42], who built a tool that can detect the description of events using *Natural Language Processing (NLP)* techniques, such as a *Position-Of-Speech (POS)* tagger, relative positional information and main entity extraction.

[3] https://taku910.github.io/mecab/

The practice of obtaining tweets for analysis that satisfy specific words is proposed by [22] for *disaster detection*. More specifically, [22] combine manual oversight with a binary text classifier to establish if an emergency-related tweet is 'relevant' or 'irrelevant'. To develop the tool, 3,785 normalised tweets are used that relate to an explosion at a power plant in Ludwigshafen, Germany in 2016. Since the tool developed by [22] requires manual oversight to realise its full potential, it may be more powerful as part of a larger and more comprehensive *disaster detection* tool, rather than as a binary text classification tool that determines if a tweet is 'relevant' or 'irrelevant'.

Another approach to *monitoring* that involves the classification of tweets is the use of spatio-temporal features located within tweets. [15] employ a three-stage approach by parsing location-related information contained in tweets. Initially, *speech tagging* and *chunking* is performed to understand the structure of each tweet. This is followed by *Named Entity Recognition (NER)* using *Conditional Random Field* from *Stanford NER* [19]. By doing this, location names contained in a tweet can be extracted and cross-referenced with a gazetteer to obtain the coordinates of locations extracted from tweets. Once complete, a multinomial *Naive Bayes* text classifier is used to identify the location and type of event detailed in a tweet.

A methodology that has been employed in other fields but only applied to *disaster detection* in 2015 is *domain adaptation classification*. [34] propose using labelled tweets from a prior source disaster, together with unlabelled tweets on a target disaster, to learn *domain adaptation classifiers*. A multinomial *Naive Bayes* text classifier and a *Bag-of-Words* model was tested on the normalised tweets that related to *Hurricane Sandy*, which affected the USA, Canada and Caribbean countries in 2012, and the *Boston Marathon Bombings*, which took place in Boston, USA in 2013. It was found that for tasks that are more specific to the new disaster being detected, *domain adaptation classifiers* perform better with closely related tweets, even if unlabelled. For tasks that are similar across disasters, classifiers that learn from source data perform better [34].

2.1.4 Hybrids

The aim of hybrid *disaster detection* tools is to leverage the strengths of one technique to compensate for the weaknesses of another. In the case of *monitoring*, [50] trained a multinomial *Naive Bayes* classifier to significantly reduce the number of tweets that are subsequently clustered. The proposed tool called *TwitterStand* is a news processing system that can detect events, such as disasters. The tool aggregates news articles taken from established sources; 2,000 hand-picked Twitter users that are known to provide reliable information in their tweets, such as links to vetted news articles that relate to an event or disaster. Similar to [50], [17] propose a hybrid tool that classifies tweets and subsequently clusters them. This involves the implementation of a *hierarchical tree structure* classifier to determine the type of disaster a tweet relates to, followed by *spatio-thematic clustering*, to estimate the geographic location of a detected disaster. Another example of this type of hybrid *monitoring* is proposed by [4], who implemented a *disaster detection* tool called *Tweedr*. The tool classifies tweets through the use of *Supervised Latent Dirichlet Allocation (sLDA)*, alongside *Support Vector Machine (SVM)* and *Logistic Regression* classifiers. The downside to the aforementioned hybrid approach is that the effectiveness of the clustering algorithms employed are highly dependent on the accuracy and the parameter values of the classifiers used to filter 'relevant' and 'irrelevant' tweets [5].

An alternative hybrid approach to *monitoring* entails clustering tweets and subsequently classifying them. This approach seeks to cluster tweets based on specified criteria, such as *relatedness* and *authority scores*. This is followed by classification, which seeks to determine the topic of the clusters formed, such as a disaster or event. [61] propose a tool called *GeoBurst*, which detects events using geotagged tweets, by identifying tweets that potentially relate to local events (called 'reference tweets') and tweets that are similar to these 'reference tweets'. Similar tweets are clustered using *pivot weighting* and *authority scores*, and subsequently classified using *wavelet analysis*. The tool proposed by [61] was tested on tweets that were posted by users located in New York and Los Angeles in the USA, since the two cities have different population distributions. Another example of a similar hybrid approach is a spatio-temporal technique proposed by [12]. This technique uses *Space Time Scan Statistics (STSS)* to cluster tweets into events, which are subsequently classified using *Latent Dirichlet Allocation (LDA)*, to determine whether the clusters detected using *STSS* relate to space-time events.

Advancing beyond the scope of Twitter, [7] developed an event detection tool that uses textual and non-textual information gleaned from social media to detect the similarity of documents. Similar to [61], this tool is not specific to *disaster detection* and is only as effective as the breadth, depth and relevance of the key words used.

Another example of a hybrid approach to *monitoring* is that proposed by [35], with their event detection tool called *Twitter-Based Event Detection and Analysis System (TEDAS)*. This approach uses online and offline stages. Offline, the tool extracts metadata and classifies tweets as relating to 'crime', a 'disaster' or 'other'. This is then fed to an online interface, where the location of crimes and disasters are estimated and ranked for real-time observation.

It could be argued that the combination of human effort and machine learning constitutes a hybrid approach. This would be on the basis of humans providing topical expertise and knowledge, where machine learning algorithms are known to perform poorly. An example of this is the *disaster detection* tool proposed by [31] for the community of Panang, Indonesia, which attempts to detect tsunamis and provide early warnings. The system presents the results from the machine learning component of the tool to analysts, allowing for different sets of needs to be catered for. Another technique that utilises human contributions with machine learning is the three-stage methodology proposed by [56]. A *learning-based* and *matching-based* technique was tested on a corpus of tweets. The *matching-based* technique seeks to improve on conventional matching systems, which simply match keywords and/or hashtags. The three-stage methodology uses a crowdsourced list of key words and hashtags to define the initial words to match.

An alternative approach to *monitoring* is one that does not assume anything about the features of an event, such as the event-type, name or location. Instead, it exploits the significant appearance of key words or temporal patterns [5]. A two-stage *disaster detection* tool is proposed by [48] that uses bigrams to analyse posts from microblogging websites, such as Twitter, in one-hour windows.

Following a similar approach to [48], who attempt to utilise the content contained in tweets and not specify specific features of a disaster, [62] propose a graphical model called *Location-Time Constrained Topic (LTT)*. This uses the content of a tweet, the time a tweet is posted and the location it was sent from. The value offered by this technique is largely based on the type of information that a user is seeking to discover. This is different to other techniques detailed in section 2.3, which details research on *situational awareness*.

2.2 Location Estimation

Following the detection of a disaster, its geographic location can be estimated. The most common approach that utilises information from Twitter is the use of coordinates contained in geotagged tweets. This information is voluntarily provided by a user and shows the precise location a tweet was sent from. The specificity of the coordinates makes this approach the most preferred for *location estimation* and arguably the most accurate. A major constraint to this approach is the lack of geotagged tweets available, since on average, only 2% of tweets worldwide are geotagged [17]. Despite the potential accuracy of using the coordinates of geotagged tweets, the approach can be skewed by posts relating to a disaster but sent from an unaffected geographic location. This is particularly problematic for 'retweets'. As a reflection of how challenging *location estimation* of a disaster is using tweets, all of the techniques outlined in the following section are hybrids.

2.2.1 Hybrids

Several simplistic approaches have been implemented that utilise a single source of location-related information from a tweet to estimate the geographic location of a disaster [2,46]. As an extension to this simplistic approach, [41] combine three location-related features embedded in a tweet using combination rules in *Dempster-Shafer theory*. These three sources are: the coordinates from recently posted geotagged tweets, the location of a user's profile and the location detailed in a tweet. The motivation for combining all three sources is because the coordinates of a recently posted tweet may quickly spread to locations that are farther away from the location of an event due to 'retweeting'. Furthermore, the absence of geotagged tweets means alternative methods should be leveraged, so the location of an event, such as a disaster, can be estimated. This can be the location attribute of a user's profile [2,46] and the extraction of location names contained within the body of a tweet [57].

The problem of not detecting events in remote places is not isolated to [41]. In the case of [24], the location of an event is estimated by matching words extracted from tweets using the *NER Ritter tool* [44] and referencing them to a gazetteer contained in the *Gate NLP framework*[4]. Despite [8] stating that the open access gazetteer used by [24] works well for detecting a location, it is a critical bottle-neck, as the proposed model is highly dependent on the gazetteer being regularly updated and accurate. An example of a hybrid approach to *location estimation* that does not perform to the standards of those detailed so far is that of [51], who implemented a text classifier to remove irrelevant tweets and cluster tweets into events using key words and *pattern matching*.

Another hybrid approach to *location estimation* is proposed by [32] with a technique called *Incremental DBSCAN*. This three-stage process dynamically weights words, clusters similar tweets and classifies events in sliding fixed periods.

2.3 Situational Awareness

The objective of *situational awareness* is to extract actionable intelligence from tweets, which can be used by emergency responders to improve the effectiveness of disaster relief operations. In order to glean pertinent information from tweets, a disaster must have been detected already. For this reason, *situational awareness* can be considered the last of the aforementioned three stages of *disaster detection*.

2.3.1 Filtering

Once a disaster has occurred, its features and characteristics can be utilised to help identify specific information. An example of this is [47], who propose an 'adaptive filter', which is a filter that adapts to the idiosyncrasies of a Twitter feed to extract disaster-related tweets. One issue with the proposed method is its reliance on high quality training data to aggregate enough relationships for accurate classification.

2.3.2 Classification

An example of a three-stage approach to extracting *situational awareness* from tweets using text classification is that of [9], who built a *situational awareness* tool for the Australian Government. Firstly, tweets are condensed into summaries, using stemmed unigrams. Secondly, tweets are classified into different types of infrastructure damage using a *SVM*. Finally, events are visually presented to watch officers. The overall purpose of the tool is to detect, assess, summarise and report messages for crisis coordination. The challenge for this approach is to adapt the model to address other types

[4] https://gate.ac.uk/

of disasters, such as floods, cyclones and bush fires [9].

A three-stage tool for *situational awareness* that implements *filtering* to significantly reduce the volume of tweets that need to be classified is proposed by [37]. Despite the tool's high Accuracy score, 95.6% of the tweets classified were not related to any event. As such, it could be argued that the dataset used to test the classifier was of a poor quality and the results should be interpreted with caution. Furthermore, the tool requires human oversight to make informed assessments of the filtered tweets. For this reason, it could be considered an effective filtering tool, rather than an effective *disaster detection* tool.

2.3.3 Hybrids

A type of *situational awareness* technique that could be considered a hybrid is one that integrates tweets and external systems. In the case of [1], the tool *Twitcident* is proposed, which relies on an external emergency broadcast service to detect a disaster. Once detected, tweets, pictures and videos relating to a disaster are aggregated from platforms, such as the now defunct *Twitpic*[5] and *Twitvid*[6]. The tool proposed by [1] is limited by the effectiveness and speed of the external emergency broadcast service in a specific country.

A similar system to [1] but applicable only for the detection of floods is proposed by [28]. The tool uses the *Global Flood Detection System (GFDS)*[7] for satellite observations of water coverage and tweets to gain a better understanding of the location, timing, causes and impacts of flooding.

Another tool that was developed specifically for the detection and tracking of floods using tweets and an official external data source, is that proposed by [14]. The tool uses authoritative data (digital elevation models, hydrological data and sensor data), combined with tweets, to perform statistical analysis and identify spatial patterns. It was found that the approach obtains the most valuable information from tweets that are sent within ten kilometres of severely flooded areas. Additionally, the tool can help emergency responders make more informed decisions during the immediate hours after a flood takes place and track the progress of a flood over time.

[26] propose a *situational awareness* tool that initially starts with a predefined *ontology*, whose categories are based on the thesis of [58]. This is based on 'gold-standard' tweets from a normalised corpus of tweets that relate to the 2011 *Joplin tornado* that hit Joplin, Missouri, USA in 2011. Out of these normalised tweets (the exact details of the normalisation process are not discussed), 'gold-standard' tweets are obtained using crowdsourced volunteers.

[5] www.twitpic.com

[6] www.twidvid.com

[7] www.gdacs.org/flooddetection/

A common issue faced by emergency responders is interpreting information gleaned from tweets. A technique that allows users to analyse large volumes of tweets on a timeline display, drill into sub-events and understand the general sentiment of tweets was proposed by [38] called *TwitInfo*. The tool requires a user to specify an event, such as a disaster, and provide supplementary information in an interactive format.

3 Disaster Detection Tool

Through the completion of the literature review, it has been identified that a computationally inexpensive automated tool that is able to *monitor* tweets, detect a disaster and estimate the geographic locations affected by a disaster would be of significant value to the humanitarian community. It is important to emphasise that this excludes *situational awareness*, therefore, the proposed tool will not attempt to glean actionable insights from tweets that would be of use to emergency responders. Figure 3 shows a conceptual overview of the proposed methodology and the techniques implemented at each stage in the proposed tool, which serve as the structure for this section. Algorithm 1 also shows the proposed methodology in the form of pseudocode.

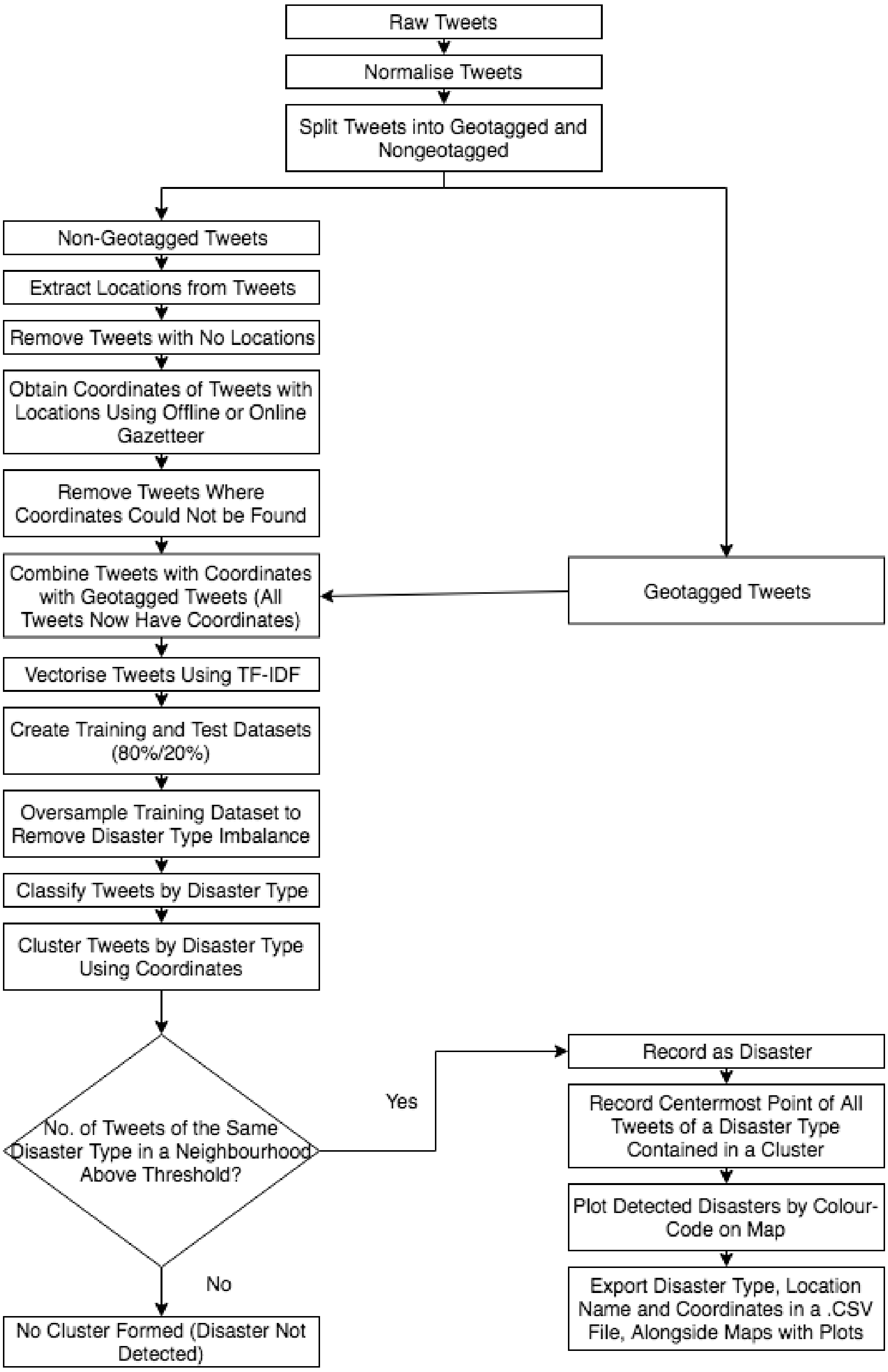

Raw Tweets
Normalise Tweets
Split Tweets into Geotagged and Nongeotagged
Non-Geotagged Tweets
Extract Locations from Tweets
Remove Tweets with No Locations
Obtain Coordinates of Tweets with Locations Using Offline or Online Gazetteer
Remove Tweets Where Coordinates Could Not be Found
Combine Tweets with Coordinates with Geotagged Tweets (All Tweets Now Have Coordinates)
Geotagged Tweets
Vectorise Tweets Using TF-IDF
Create Training and Test Datasets (80%/20%)
Oversample Training Dataset to Remove Disaster Type Imbalance
Classify Tweets by Disaster Type
Cluster Tweets by Disaster Type Using Coordinates
No. of Tweets of the Same Disaster Type in a Neighbourhood Above Threshold?
Yes
Record as Disaster
Record Centermost Point of All Tweets of a Disaster Type Contained in a Cluster
Plot Detected Disasters by Colour-Code on Map
Export Disaster Type, Location Name and Coordinates in a .CSV File, Alongside Maps with Plots
No
No Cluster Formed (Disaster Not Detected)

Fig. 3: Workflow of the proposed automated *disaster detection* tool (not ‘testing’ or ‘evaluation’)

Algorithm 1 shows the pseudocode for the proposed tool, outlining the methodology employed at each stage:

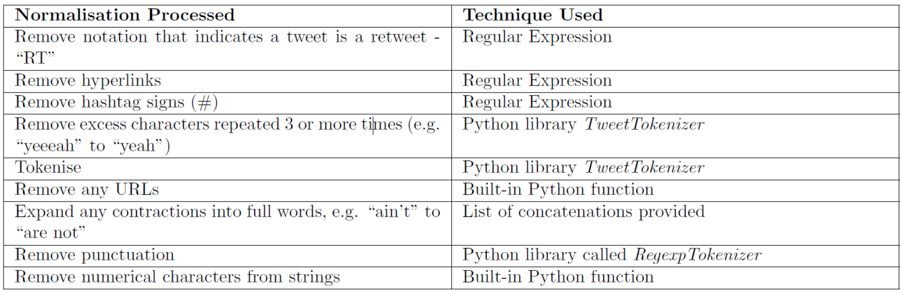
```
INPUT: RAW TWEETS
begin
    (1) NORMALISE TWEETS
    (2) SPLIT TWEETS INTO GEOTAGGED AND NON-GEOTAGGED TWEETS
    if Tweet has coordinates appended then
        Place in a separate dataframe for geotagged tweets
    else
        Place in a separate dataframe for non-geotagged tweets
    end
    (3) IDENTIFY LOCATION AND COORDINATES OF NON-GEOTAGGED TWEETS
    if A tweet contains the name of a geographic location then
        Return coordinates from an offline gazetteer
    else
        Return coordinates from an online gazetteer
    end
    if No coordinates found for location (city or country) then
        Discard tweet
    end
    if Coordinates of a city and country exist then
        Use the city location and coordinates
    else
        Use the country location and coordinates
    end
    (4) COMBINE TWEETS WITH COORDINATES WITH GEOTAGGED TWEETS
    (5) VECTORISE TWEETS USING TF-IDF
    (6) CREATE TRAINING & TEST DATASETS
    (7) OVER-SAMPLE TRAINING DATASET, IF NECESSARY
    Synthetic minority over-sampling (SMOTE) was applied on the dataset used
    (8) CLASSIFY TWEETS BY DISASTER-TYPE
    Fit selected classifier to over-sampled training data
    Make predictions for test data using the fitted classifier
    Export the predictions for each disaster-type as separate datasets
    (9) ESTIMATE LOCATION AND CLUSTERING TWEETS BY DISASTER-TYPE
    for $DisasterType \in DisasterTypes$ do
        if Number of tweets > density threshold specified then
            Cluster is formed, which is a disaster
            Centermost point of the cluster (disaster), is its estimated location
        else
            No cluster formed (no disaster detected)
        end
    end
    (10) PLOT AND RECORD EACH DISASTER FOR EACH DISASTER-TYPE
    for $DisasterType \in DisasterTypes$ do
        for $Disaster \in Disasters$ do
            Plot the estimated location of each disaster on a world map
            Add the estimated location to a dataframe to be exported as a .csv
        end

    end
end
OUTPUT: DISASTER-TYPE, LOCATION NAME AND COORDINATES
```

Algorithm 1: Pseudocode for the proposed automated *disaster detection* tool

3.1 Normalise Tweets

The process of text normalisation entails standardising text. This is particularly important for documents, such as tweets, which are highly informal. It is stated by [6] that text normalisation is often incorrectly implemented in a standardised manner, irrespective of the type of document being normalised. The following normalisation techniques were performed for each raw tweet in the corpus used. The decision of what techniques to implement was based on manual testing and observations on the dataset [27].



Normalisation Processed	**Technique Used**
Remove notation that indicates a tweet is a retweet - "RT"	Regular Expression
Remove hyperlinks	Regular Expression
Remove hashtag signs (#)	Regular Expression
Remove excess characters repeated 3 or more times (e.g. "yeeeah" to "yeah")	Python library *TweetTokenizer*
Tokenise	Python library *TweetTokenizer*
Remove any URLs	Built-in Python function
Expand any contractions into full words, e.g. "ain't" to "are not"	List of concatenations provided
Remove punctuation	Python library called *RegexpTokenizer*
Remove numerical characters from strings	Built-in Python function

Fig. 4: Normalisation process and techniques used

In addition to the text normalisation techniques, the following techniques were explored:

- *Slang*: converting slang words into synonyms that can be identified in an English dictionary
- *Lemmatisation*: the root of a word is obtained, which is called a "lemma"
- *Stemming*: the end of a word is removed that has derivational affixes
- *Case folding*: make all words in a document lowercase
- *Stop-words*: remove frequently used words in the English language

It was decided not to correct slang, as manual observations of the corpus showed little slang being used, making the correction on a large corpus computationally wasteful. Both lemmatisation and stemming were found to reduce the meaning of a tweet, hence their exclusion. The case folding of tweets was not performed, since the identification of geographic locations detailed within tweets is more robust when the names of locations (often capitalised) are left unchanged. Lastly, stop-words were not removed during normalisation, as they are treated using an information retrieval technique called *Term Frequency - Inverse Document Frequency (TF-IDF)* [54] that is implemented before classification in the proposed tool. *TF-IDF* generates a numerical value for each word, which indicates its importance to a document, such as a tweet. This value increases proportionally to the frequency of a word in a tweet but is offset by the frequency of the word in the corpus of tweets. As such, this approach adjusts for generic words that appear frequently. After text normalisation has been performed, any tweets that contain less than four words are removed from the final list of processed tweets, to prevent the needless processing of tweets that are of no value for text classification. This is consistent with [18], who only retain tweets for sentiment analysis that contain more than three words. The effect of normalisation on the number of tweets that were omitted from analysis due to their final tweet length being less than four words can be seen in figure 5, which shows the percentage of original tweets remaining after text normalisation was performed:

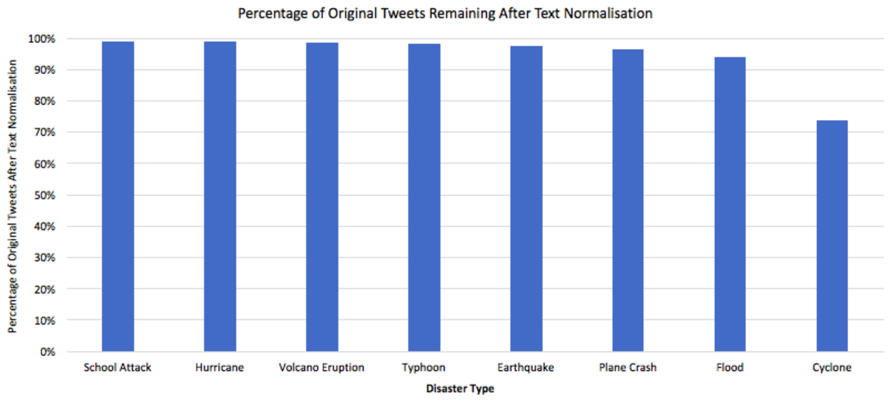


Fig. 5: Percentage of tweets per disaster type retained after text normalisation

The overall purpose of implementing text normalisation in the proposed *disaster detection* tool is to enable the text contained in tweets to be accurately processed by *NLP* techniques, such as *NER*, to establish if any geographic locations are referenced in the body of a tweet that is not geotagged. Text normalisation also enables the text classifiers used to identify relationships and important features that can help accurately determine the disaster-type a tweet relates to.

3.2 Split Geotagged and Non-Geotagged Tweets

To determine the geographic location of a disaster, the geographic location of where a tweet was sent from must be estimated. One of the most reliable and frequently used sources of information to estimate the geographic location of a disaster is via the location a tweet was sent from. If a tweet is geotagged by a user, the coordinates are available. If, however, a tweet is not geotagged, an alternative source of information that can be used is the name of locations detailed in the body of a tweet. As such, stage three (identifying the name of a location and obtaining its coordinates) is only performed for tweets that are not geotagged. To do this, tweets that are not geotagged are separated from those that are, before commencing with stage three.

3.3 Identify Location and Coordinates of Non-Geotagged Tweets

For all tweets that are not geotagged, a Python library called *GeoText*[8] is used to extract the names of cities and countries mentioned in the body of a tweet. If a city is

[8] https://pypi.org/project/geotext/

identified, the country that the city is located in is also returned. If only a country is identified, no city is returned, since only a country is mentioned in the text. If no location is identified by *GeoText* in a tweet, the tweet is discarded. If a location is identified, an attempt is made to obtain the coordinates of the location using a gazetteer. Due to the high volume of tweets that are processed, an offline gazetteer is referenced first. The database that forms the offline gazetteer was provided by *GeoNames*[9]. This initially contained over ten million geographical names but it was reduced to 7.2 million entries after removing duplicates. Using the offline gazetteer enabled the names of geographical locations to be converted into coordinates. If this is unsuccessful, an online gazetteer, in the form of a Python library called *GeoPy*[10] is used.

To minimise the computational cost of obtaining the coordinates of locations identified in a tweet, all locations found in tweets that are not geotagged are converted to a list. All duplicate entries in the list are then removed to prevent searching for coordinates of the same location more than once. The coordinates relating to the location names in this list containing no duplicates are first identified in the offline gazetteer and if this process is unsuccessful, the online gazetteer is used. In either case, any coordinates found are appended to dictionaries created for cities and countries (one each). The updated dictionary subsequently becomes the first object to be searched when attempting to identify the coordinates of each location in the list that contains no duplicates (i.e. before the offline and online gazetteers). Once all coordinates have been located, the original list of locations is then iterated through (potentially containing the same location multiple times, if the location is contained in several tweets) and the coordinates taken from their respective dictionaries. By creating a list of locations, removing duplicates, storing any coordinates found in a dictionary and checking the dictionary before searching in a gazetteer, the execution time of the process is significantly reduced.

Since the primary source of information used for *location estimation* are the coordinates of where a disaster-related tweet was sent from, cities are prioritised over countries, since they are more precise in their geographic location. As such, only one location name and its respective coordinates are stored for each tweet, with cities taking priority over countries. Furthermore, whenever more than a single city or country is detected in a tweet by *GeoText*, only the first is used. By doing this, if a tweet contains a location name that is located in either of the gazetteers used, the most accurate coordinates are assigned to the tweet. It is recognised that this approach is simple in nature but it is assumed that the most important location will be detailed first in a tweet.

[9] http://download.geonames.org/export/dump/

[10] https://geopy.readthedocs.io/en/stable/

3.4 Combine Tweets With Coordinates With Geotagged Tweets

Now that the location name and its respective coordinates have been assigned to every tweet that contains the name of a location in its body and has had its coordinates identified, the previously non-geotagged tweets that now have a location name and coordinates identified are joined with the geotagged tweets to form one dataframe. These tweets are now ready to be classified by disaster-type in stage five.

3.5 Vectorise Tweets Using TF-IDF

TF-IDF is implemented, which is a numerical value that indicates how important a word is to a document, such as a tweet. This value increases proportionally to the frequency of a word in a tweet but offset by the frequency of the word in the corpus of tweets. As such, this approach adjusts for generic words that appear frequently. Due to its significant processing cost, the vocabulary size is limited to 10,000 words. Efforts to increase the vocabulary size were prevented by the computational cost required. During 'testing', the performance of the four classifiers is based on *TF-IDF* vectors produced for unigrams, bigrams and trigrams. Based on the performances of the two selected classifiers, one of the n-grams is selected for the *TF-IDF* vector used for 'evaluation'.

3.6 Create Training and Test Datasets

After a *TF-IDF* vector has been produced, the corpus of tweets is split into training and test datasets, with a 80%/20% split, respectively.

3.7 Over-sample Training Dataset

Synthetic Minority Over-Sampling Technique (SMOTE) [11] is used on the training dataset only, to correct the large class imbalance. *SMOTE* over-samples the minority class creating 'synthetic' examples, rather than by over-sampling with replacement. The test set remains untouched, since the volume of tweets is used in stage nine to cluster tweets and determine if a disaster has taken place.

3.8 Classify Tweets by Disaster-Type

In the corpus of tweets used, there is a total of eight different disaster-types (classes). During 'testing' and 'evaluation', the selected classifiers are set to *One-Vs-Rest*, also referred to as *One-Vs-All* or *One-Against-All*, using the *sklearn.multiclass*[11] library. Given *N* classes, the *One-Vs-Rest* approach trains *N* different binary classifiers. Each of the binary classifiers is trained on examples of a single class and all remaining

[11] http://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html

classes [43]. The dataset used in this paper has eight disaster types. This means that eight binary classifiers are trained on the training dataset. A test example (a tweet), is then fed to each of the eight binary classifiers that have been trained, each predicting a class membership probability. The test example (a tweet) is assigned to the class with the highest probability. The input to each model is a *TF-IDF* vector. In the case of 'testing', *TF-IDF* vectors are produced for unigrams, bigrams and trigrams (stage five of the methodology). For 'evaluation', the n-gram variation that enables the most successful classification to be performed will be selected and a *TF-IDF* vector created, accordingly (this is established from the results obtained during 'testing'). Four classification algorithms are benchmarked; *Naive Bayes*, *Logistic Regression*, *Support Vector Machine* and a *Random Forest*.

Once the four text classifiers have been assessed in the 'testing' phase (the full details of 'testing' and the results obtained are detailed in section 5.2), two classifiers are selected for 'evaluation'. The first is the best performing classifier, which has optimised parameter values. The second is a *Naïve Bayes* classifier, which has optimised parameter values. The *Naïve Bayes* classifier is selected so that it can be compared to the best performing classifier. This is appropriate given its simplicity and its relatively low computation cost, in comparison to the other classifiers tested. The details and results from the 'evaluation' phase are detailed in 5.3.

In the proposed *disaster detection* tool, once all tweets have been classified by disaster-type using the selected text classifier, the tweets are subsequently grouped by disaster-type and segregated into separate dataframes. This allows clustering to be performed by disaster-type using coordinates, enabling multiple disaster-types to be detected in the same geographical location.

3.9 Cluster Tweets by Disaster-Type Using Coordinates

In the literature review conducted in section 2, it was identified that *DBSCAN* is the state-of-the-art clustering technique for *disaster detection*. As aforementioned, for each tweet in a cluster, the neighbourhood of a specified radius must contain a minimum number of tweets to form a unit of density region. This approach treats outliers as noise, makes no assumption on the number of clusters and is capable of identifying randomly-shaped clusters. For this reason, a *DBSCAN* algorithm is implemented in the proposed *disaster detection* tool, for the purpose of detecting a disaster using the coordinates of classified tweets.

When implementing *DBSCAN*, there are two key parameter values that need to be specified: *epsilon* (henceforth referred to as *eps*) and *minimum points* (henceforth referred to as *min_points*). The *eps* value determines the neighbourhood radius around a point. The *min_points* value is the fewest number of points within an *eps* value required to form a density cluster.

It is suggested by [60] that there is no general solution to determining the appropriate values for the parameters *min_points* and *eps*. For example, a *K-nearest neighbour* query becomes too computationally expensive on a very large database. Furthermore, limiting the sample size to overcome this will significantly decrease the accuracy of the estimation made [60]. In the case of [10], the *eps* value was set to 250 meters and the *min_points* value set to ten (number of tweets). These values are low in nature, since the events being identified are sub-events in the city of Barcelona, Spain. In the case of [32], an *eps* value of 0.4 miles and a *min_points* value of 15 was used, however, these low values were chosen as they are applicable to each sliding time window, rather than a large volume of tweets, irrespective of when they were posted or processed. Using these two papers as proxies, three different *eps* values and three different *min_points* values will be tested. These are 20km, 40km and 60km for *eps* and 20, 50 and 80 tweets for *min_points*. In comparison to [10] and [32], these larger values reflect a larger corpus being used, the global geographic range of the corpus content and the static time horizon of the tweets, as opposed to the sliding windows implemented by [32].

To implement *DBSCAN*, a *ball-tree* algorithm is selected [40], which is efficient on highly-structured data, such as the coordinates of tweets, which follow the format of (latitude, longitude). The *Haversine distance* is used to calculate the *Great-circle distance*, which is the shortest distance between two points across the surface of a sphere.

3.10 Plot and Record Each Disaster for Each Disaster-Type

If a cluster is formed, equating to a disaster being detected, the centermost point for all tweets forming the cluster is calculated and used as the estimated location of the disaster. Each disaster is plotted on a map of the world using colour-coordinated dots, with each colour representing a different disaster-type. An example of this can be seen in figure 6 below:

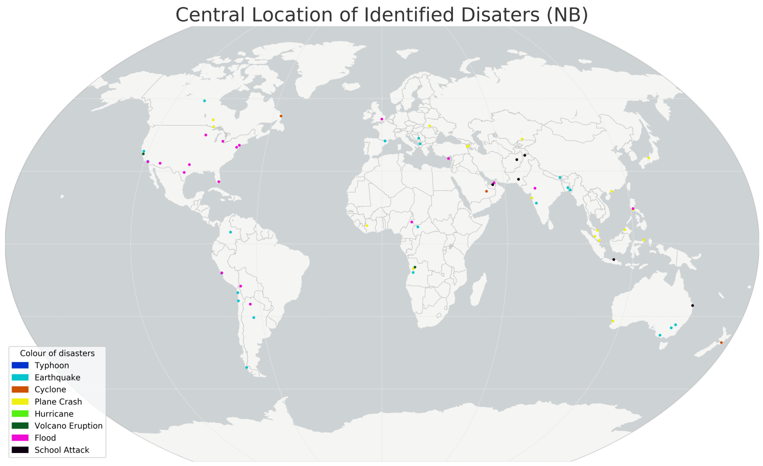


Fig. 6: Example of a plot generated by the *disaster detection* tool. Each colour-coded dot is an estimated location of a disaster

Since figure 6 only gives users a macro-view of locations affected by a disaster, the name and coordinates of the estimated location(s) of each disaster are also recorded. Details of the outputs from the automated *disaster detection* tool are detailed in the following stage.

3.11 Output

Once the centermost point for all tweets forming a cluster is calculated and used as the estimated location of the disaster, the plot created in stage ten is exported as a .png image, with each disaster-type assigned a specific colour. In addition to this, a .csv file is exported, which contains the disaster-type, location name and coordinates of the detected disaster, which can be subsequently used for other purposes, such as *situational awareness*.

3.12 Software

The proposed *disaster detection* tool was built in Python 3.6. This enables the entire script to run uninterrupted with full automation. The reason for using Python for this task is because of the extensive number of libraries that facilitate *disaster detection*. Examples include *GeoText* to perform *NER* for the names of geographic locations and *GeoPy*, to identify the coordinates of location names.

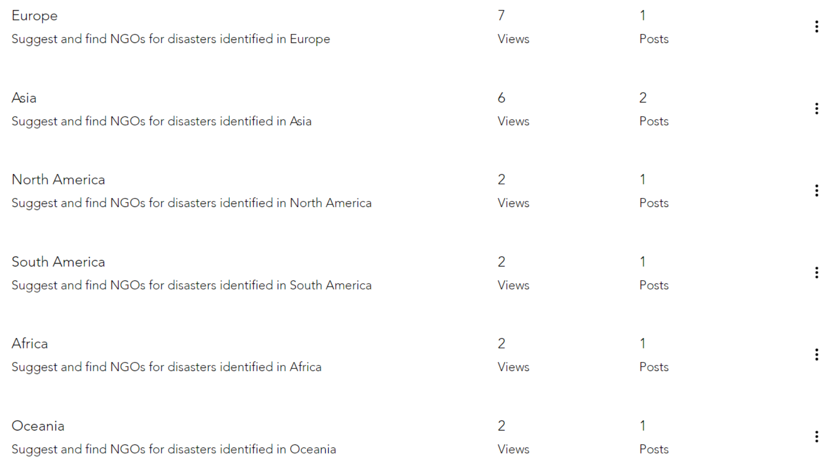
4 Crowdsourcing Portal for NGO Identification

In a study performed by [20], it was found that the number of NGOs within a country has no significant impact on the disaster response costs. This directly contradicts the intuition that the increased presence of NGOs reduces the level of external support required to provide effective disaster relief. Two reasons provided by [20] for this are: (1) some events may be beyond the capabilities of some NGOs, or (2) NGOs do not communicate well among themselves to deliver a coordinated response with the international community. In both of these instances, it could be argued that the identification of local and regional NGOs who are capable of delivering support to a specific disaster would help increase the overall effectiveness of the disaster relief support. This is due to NGOs being able to focus on their strengths, which in turn could lead to the international community being able to provide support where local NGOs require it.

The following section shows the structure of the proposed portal and how individuals can exchange and critique NGO suggestions.

4.1 Structure of the Crowdsourcing Portal

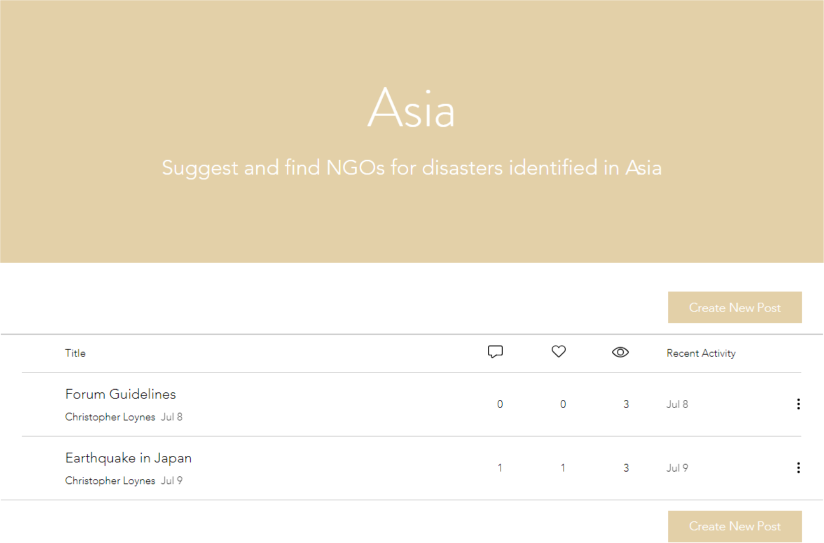
Figure 7 shows the structure of the proposed portal, which is segregated by region:



Europe Suggest and find NGOs for disasters identified in Europe	7 Views	1 Posts	⋮
Asia Suggest and find NGOs for disasters identified in Asia	6 Views	2 Posts	⋮
North America Suggest and find NGOs for disasters identified in North America	2 Views	1 Posts	⋮
South America Suggest and find NGOs for disasters identified in South America	2 Views	1 Posts	⋮
Africa Suggest and find NGOs for disasters identified in Africa	2 Views	1 Posts	⋮
Oceania Suggest and find NGOs for disasters identified in Oceania	2 Views	1 Posts	⋮

Fig. 7: Structure of the proposed portal, which is categorised by geographic region

Once a region is selected, a user can post a suggestion, which in turn can be critiqued by other users via replies. The structure of a region is shown in figure 8:



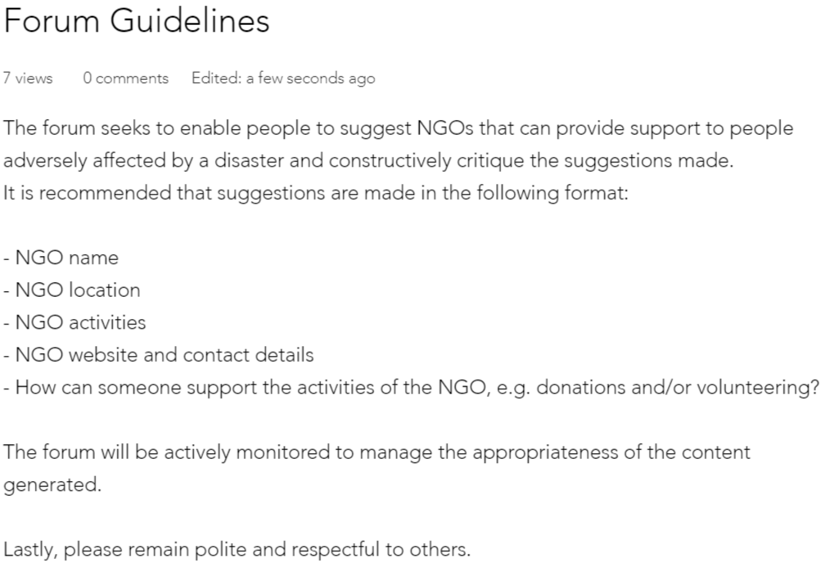
Asia

Suggest and find NGOs for disasters identified in Asia

Title	💬	♡	👁	Recent Activity
Forum Guidelines Christopher Loynes Jul 8	0	0	3	Jul 8
Earthquake in Japan Christopher Loynes Jul 9	1	1	3	Jul 9

Fig. 8: Structure of a region listed on the portal. The example focuses on Asia, with a test post titled ‘Earthquake in Japan’

A template suggestion of how a post on the portal should be structured is shown in figure 9:



Forum Guidelines

7 views 0 comments Edited: a few seconds ago

The forum seeks to enable people to suggest NGOs that can provide support to people adversely affected by a disaster and constructively critique the suggestions made.
It is recommended that suggestions are made in the following format:

- NGO name
- NGO location
- NGO activities
- NGO website and contact details
- How can someone support the activities of the NGO, e.g. donations and/or volunteering?

The forum will be actively monitored to manage the appropriateness of the content generated.

Lastly, please remain polite and respectful to others.

Fig. 9: A template of how users should structure their suggestions on the portal

If users follow the structure outlined in figure 9 to post their suggestions and administrators of the website review posts regularly, the portal has the potential to act as an efficient mechanism to exchange information and identify local and regional NGOs that can provide disaster relief to individuals adversely affected by a disaster.

4.2 NGO Suggestion Scoring System

In order to differentiate the value of suggestions made by users on the portal, a scoring system is proposed. The scoring system seeks to leverage the proposed structure of the post template detailed in figure 9. The scoring system ranges from zero to five (inclusive) and is awarded by an administrator to a suggestion made. Each of the following points that are verified by three other users on the portal can receive a score of 0, 0.5 or 1:

- NGO name and location
- NGO activities
- NGO website
- NGO contact details
- What types of payment can be used to make donations to the stated NGO

The process of verification entails other users agreeing with the evidence provided. It is recognised that some local NGOs in remote locations may not be as visible as other NGOs. As such, the level of evidence available to new or lesser-known NGOs may be difficult to obtain. This is where the interpretation from an administrator is required. A score of 0 is awarded to any point that has not been verified by three other users. A score of 0.5 is awarded if more than three users agree with a point made (e.g. NGO contact details) but some conflicting information is outlined by other users. Lastly, a score of 1 is awarded if more than three users verify a point stated in the suggestion and there is no conflicting information provided by others. These individual scores are aggregated and posted in the title of a suggestion for other users to observe. After being awarded a score, suggestions will be displayed in their relevant sections in descending order, based on the score awarded by an administrator. It is recognised that the proposed approach is manually intensive, however, the harnessing of crowdsourced information in the context of NGO identification involves qualitative data and as such, requires quality assessment, which is provided by the administrators on the portal.

5 Empirical Investigation and Findings

In this section, we illustrate the effectiveness of the disaster detection tool on various corpora generated from disaster feeds.

5.1 Datasets Used

A corpus of labelled tweets is used that is provided by [27]. This corpus covers the following man-made and natural disasters:

1. **Earthquake** - Nepal (2015). California, USA (2014). Iquique, Chile (2014)
2. **Typhoon** - Vietnam (2014). Philippines (2014)
3. **School attack** - Peshawar, Pakistan (2014)
4. **Cyclone** - South Pacific Ocean (2015)
5. **Plane crash** - Flight MH370 (2014)
6. **Hurricane** - California, USA. Mexico (2014)
7. **Volcano eruption** - Baroarbunga, Iceland (2014)
8. **Flood** - India (2014). Pakistan (2014)

The original dataset provided by [27] contains 12,021,227 tweets. Figure 10 shows the quantity of tweets in the original corpus for each of the eight disaster-types:

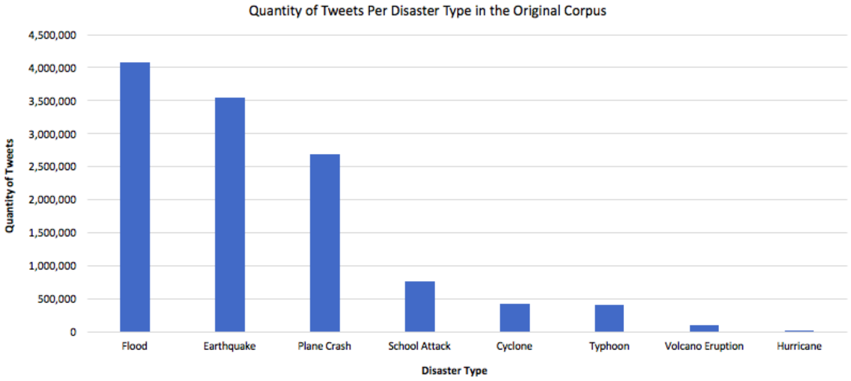


Fig. 10: Quantity of tweets per disaster-type in the original corpus provided by [27]

Due to the processing power required to process all of the tweets in the original corpus, a sample of 299,910 tweets was randomly selected, equating to approximately 2.5% of the original corpus. This is performed after text normalisation. The percentage of tweets retained after random sampling per disaster-type is shown in figure 11:

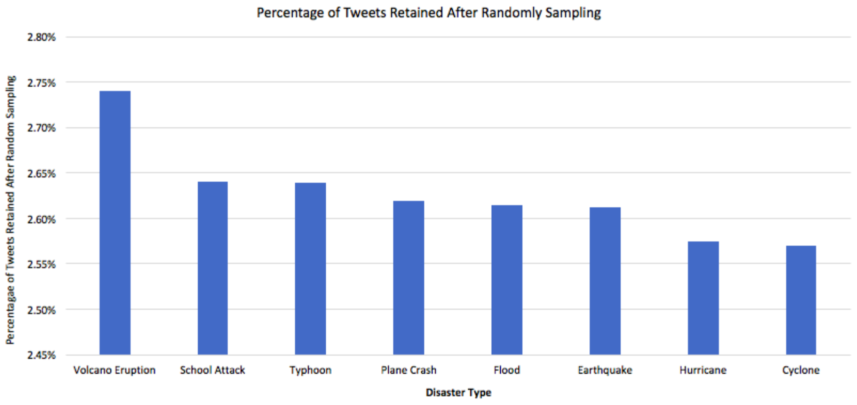


Fig. 11: Percentage of tweets retained after randomly sampling

Figure 11 shows that the random sampling has retained a very similar percentage of the original tweets for each disaster-type, which preserves the composition of the original corpus. More importantly, the reduced dataset allows for the different components of the proposed tool to be tested and the overall tool to be appropriately evaluated, given the processing power that was available, which was a 2.3GHz Intel Xeon E5 processor with 64GB of RAM, rented from Amazon Web Services (AWS)[12].

5.1.1 Sample Bias Correction

As shown in figure 10, the corpus of tweets used in this paper is heavily imbalanced, with an overwhelming majority of tweets relating to the disaster-types ‘flood’, ‘earthquake’ and ‘plane crash’. This class imbalance has been dealt with differently during different stages of developing the proposed tool. Overall, two stages of analysis are performed to select a text classifier. The first is ‘testing’, which analyses four text classifiers, of which a benchmark classifier and the best performing classifier are selected. These classifiers are then analysed in detail in the second stage of analysis called ‘evaluation’. During ‘testing’, the four text classifiers are tested using *K-fold stratified sampling* at the same time as *ten-fold cross validation*. This was due to the practical efficiency offered by the Python library *cross_val_predict*. During ‘evaluation’, the performance of the benchmark and the best performing classifier are assessed on five increasingly larger randomly sampled sub-sets (20%, 40%, 60%, 80% and 100% of 299,910 normalised tweets) to simulate the effects of streaming tweets, in an attempt to understand how it affects each classifier’s performance. Since *ten-fold cross validation* was too computationally expensive to perform on the full 299,910 normalised tweets during ‘evaluation’, the two text classifiers are fitted using training data that is over-sampled with *Synthetic Minority Over-Sampling Technique (SMOTE)* [11], to deal with the class imbalance. *SMOTE* over-samples the minority class creating ‘synthetic’ examples, rather than by over-sampling with replacement. More specifically, minority classes are over-sampled by taking a sample

[12] https://aws.amazon.com/

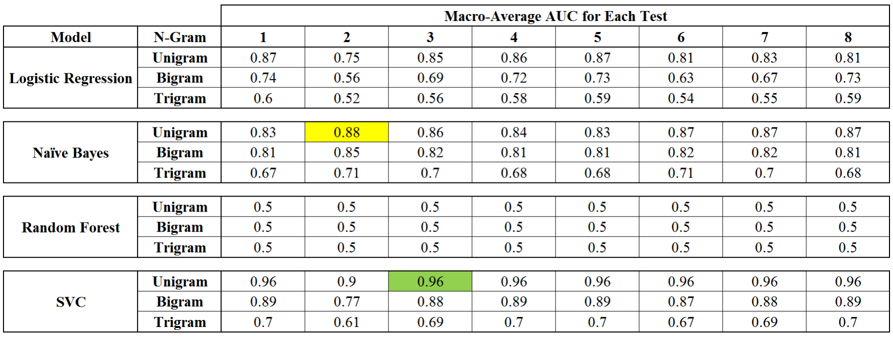
from each minority class and introducing synthetic examples along the line segments joining any/all of the K minority class nearest neighbours [11]. For completeness, the over-sampling technique *Adaptive Synthetic Sampling Approach (ADASYN)* [23] was explored but it was deemed too computationally expensive.

It is important to note that during 'evaluation', over-sampling was not performed on the test dataset and only on the training dataset. This was a conscious decision, as the number of tweets that belong to a disaster-type and have coordinates within a specified neighbourhood will determine if a disaster is detected, as they are parameters in the *DBSCAN* algorithm that is implemented and discussed in section 5.4. Consequently, a potentially imbalanced test dataset is not over-sampled, in order to preserve the integrity of the data and not artificially create a disaster by over-sampling tweets in the test dataset.

5.2 Disaster Detection Testing Results

During 'testing', four multinomial text classifiers are tested; a *Naive Bayes*, *Logistic Regression*, linear *Support Vector Classifier (SVC)* and *Random Forest*. In this paper, a linear kernel is selected for the *SVC*, as it is suggested by [25] that it is the most appropriate kernel for text classification, since most text is linearly separable. All classifiers are tested on 100,138 normalised tweets provided by [27]. The *cost* parameter settings for the *SVC* and *Logistic Regression* classifiers are adjusted per test. For the *Naive Bayes* classifier, the *alpha* value is adjusted, which is a smoothing parameter. A value of zero means no smoothing is performed. Eight different values, between 0 and 1, inclusive, are tested for each of the parameters, in addition to unigrams, bigrams and trigrams, to determine what the most appropriate values are to optimise the performance of the classifiers. The parameter adjustments made for the *Random Forest* classifier tested includes: the number of trees in the forest, the maximum depth of a tree and the splitting criterion (*Gini Impurity* and *Entropy*). During 'testing', each of the four classifiers use *ten-fold cross validation*, combined with *K-fold stratified sampling*, which returns stratified folds, preserving the percentage of samples for each class. Consequently, the analysis of each classifier's performance during 'testing' focuses on macro-average metrics, rather than micro-averaged metrics, as macro-averaging treats all classes equally and micro-averaging favours bigger classes [53].

The macro-average *Area Under Curve (AUC)* scores, which is the area under a *Receiving Operating Characteristic (ROC)* curve, for each classifier, for each n-gram test and for each parameter value adjustment, can be seen in figure 12. The test generating the highest score for the benchmark *Naive Bayes* classifier and the highest scoring classifier overall is highlighted yellow and green, respectively:



		Macro-Average AUC for Each Test							
Model	**N-Gram**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Logistic Regression	**Unigram**	0.87	0.75	0.85	0.86	0.87	0.81	0.83	0.81
	Bigram	0.74	0.56	0.69	0.72	0.73	0.63	0.67	0.73
	Trigram	0.6	0.52	0.56	0.58	0.59	0.54	0.55	0.59
Naïve Bayes	**Unigram**	0.83	0.88	0.86	0.84	0.83	0.87	0.87	0.87
	Bigram	0.81	0.85	0.82	0.81	0.81	0.82	0.82	0.81
	Trigram	0.67	0.71	0.7	0.68	0.68	0.71	0.7	0.68
Random Forest	**Unigram**	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Bigram	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Trigram	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
SVC	**Unigram**	0.96	0.9	0.96	0.96	0.96	0.96	0.96	0.96
	Bigram	0.89	0.77	0.88	0.89	0.89	0.87	0.88	0.89
	Trigram	0.7	0.61	0.69	0.7	0.7	0.67	0.69	0.7

Fig. 12: AUC scores: different n-grams and parameter values

Figure 12 shows little variability in the AUC scores when the parameter values are adjusted. The highest AUC score is 0.96 was obtained by the *SVC* during test three for unigrams, which corresponds to a *cost* value of 0.5. In the case of the *Naive Bayes*, the highest scoring AUC value is 0.88, which was obtained in test two, with an *alpha* value of 0.1. The AUC scores for the *Random Forest* classifier remain unchanged at 0.5 across all n-gram variations and parameter value adjustments. A suggestion for the lack of variability is that the algorithm takes a consensus of 'votes' from the decision trees generated. As such, across 100 and 200 decision trees, little variability is achieved. This view is supported by [16], who state that changes in the parameter values for a *Random Forest* classifier have negligible effects in most cases.

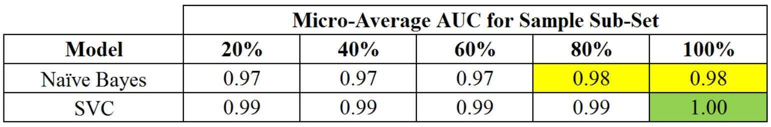
Other than the *Random Forest* classifier, the introduction of bigrams and trigrams caused a deterioration in the AUC scores. This is consistent with [22], who found that the use of n-grams deteriorated text classifier performances. This may be explained by the classifiers over-fitting when longer n-grams are added, despite implementing *Term Frequency - Inverse Document Frequency (TF-IDF)* [54]. Similar to the AUC scores, the *SVC* achieved the highest Accuracy score, with a value of 0.9649 for unigrams during test three. For the *Naive Bayes*, this was highest in test eight for unigrams. Consistent with the AUC scores, the Accuracy scores of all classifiers tested deteriorated for bigrams and trigrams, compared to unigrams. Taking this and the AUC scores into account, the linear *SVC* classifier with a *cost* value of 0.5 (test three) and the *Naive Bayes* classifier with an *alpha* value of 0.1 (test two) are selected for 'evaluation'. 'Evaluation' is where the *Naive Bayes* and linear *SVC* classifiers are tested on five increasingly larger randomly sampled sub-sets of tweets to simulate the effect of streaming tweets and a single classifier is selected for the proposed *disaster detection* tool.

5.3 Text Classification Evaluation Results

After selecting the linear *SVC* with a *cost* value of 0.5 and the *Naive Bayes* classifier with an *alpha* value of 0.1, both classifiers are evaluated on increasingly larger sample sub-sets of the corpus. Specifically, five sub-sets are randomly selected: 20%, 40%, 60%, 80% and 100% of a corpus comprised of 299,910 normalised tweets. This is to determine if the number of tweets processed impact the performance of the text classifiers.

Both classifiers are evaluated by generating ROC curves and micro-average metrics. Since no over-sampling was performed to the test dataset, a class imbalance exists. For this reason, the selection of the appropriate text classifier is based on micro-average metrics and not macro-average metrics.

5.3.1 Micro-Average AUC Scores



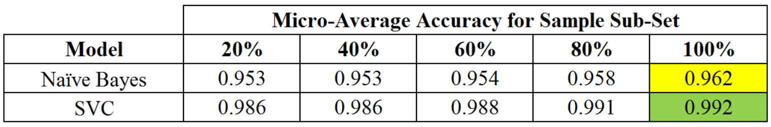
	Micro-Average AUC for Sample Sub-Set				
Model	**20%**	**40%**	**60%**	**80%**	**100%**
Naïve Bayes	0.97	0.97	0.97	0.98	0.98
SVC	0.99	0.99	0.99	0.99	1.00

Fig. 13: AUC scores for each sample sub-set evaluated

All AUC scores in figure 13 are close to one. This strongly suggests that the text normalisation procedure and implementation of *TF-IDF* are both effective. In addition to this, a manual observation of the corpus of tweets identified that the language contained in the tweets is very explicit. Little ‘noise’ exists that causes the text classifiers to incorrectly classify tweets. In figure 13, a subtle increase can be observed for both text classifiers when the sample sub-set size increases. This could indicate that the performance of the text classifiers improves as the training dataset of normalised tweets used to fit the classifiers increases.

5.3.2 Micro-Average Accuracy/Precision/Recall/F-Scores

The Accuracy scores for both classifiers all exceed 95%. Overall, the linear *SVC* performs best, with a top score of 0.992 for a 100% sample sub-set, with the *Naive Bayes* classifier scoring 0.962. Similar to the AUC scores, the performance of both classifiers improves as the sample sub-set size increases. Since the micro-average of each metric computed aggregates the contributions of all classes, the Precision, Recall and F-scores are all identical to the Accuracy scores detailed in figure 14. Across all metrics, the linear *SVC* performs best.



	Micro-Average Accuracy for Sample Sub-Set				
Model	**20%**	**40%**	**60%**	**80%**	**100%**
Naïve Bayes	0.953	0.953	0.954	0.958	0.962
SVC	0.986	0.986	0.988	0.991	0.992

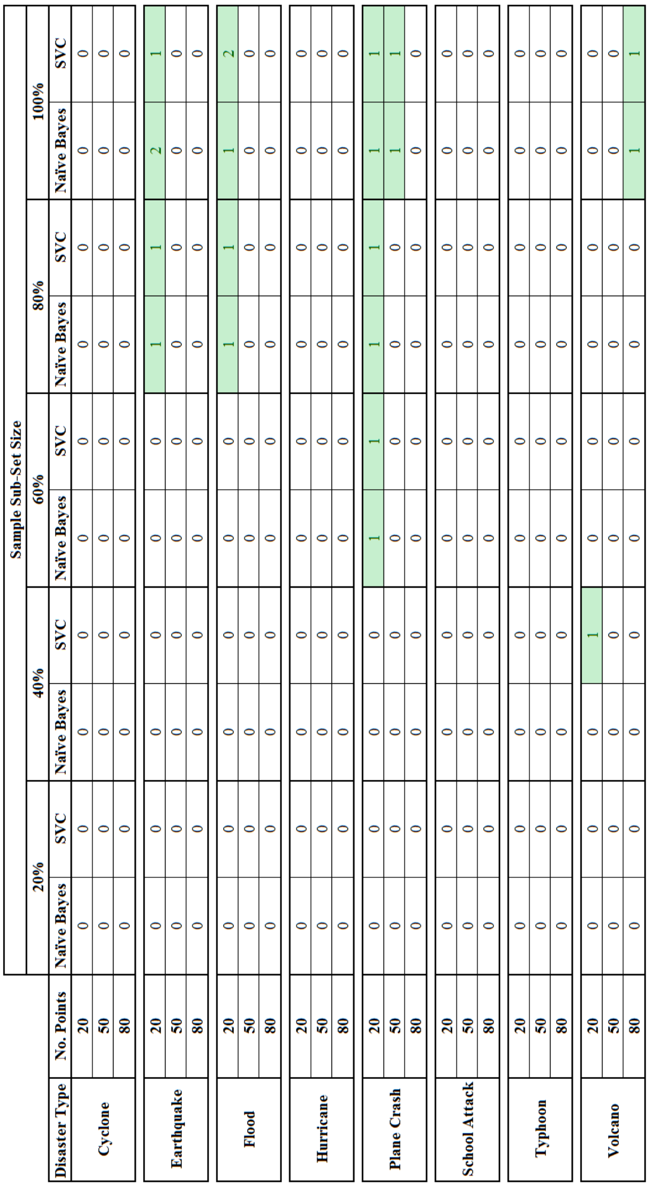
Fig. 14: Accuracy Scores for each sample sub-set evaluated

5.4 Clustering Results

Following classification, each tweet is processed using *Density-based spatial clustering of applications with noise (DBSCAN)*. Two key parameters are adjusted: *eps* (the neighbourhood radius around a point) and *min_points* (the minimum density in a neighbourhood to form a cluster). Each cluster is treated as a disaster, with each point representing a tweet's coordinates.

In order to establish the most appropriate values for the two parameters, their values are adjusted and the results analysed. The *eps* value is adjusted to 20km, 40km and 60km. The *min_points* value is adjusted to 20, 50 and 80. These values are based on [10] and [32] but adjusted to suit the global nature and scale of the corpus tested. Both the *Naive Bayes* and linear *SVC* classified tweets are used to test the *DBSCAN* algorithm, to understand the impact it has on its performance. Since increasingly larger and randomly sampled sub-sets are used to test the performance of the *DBSCAN* algorithm, the number of tweets belonging to each disaster-type differs in each test performed.

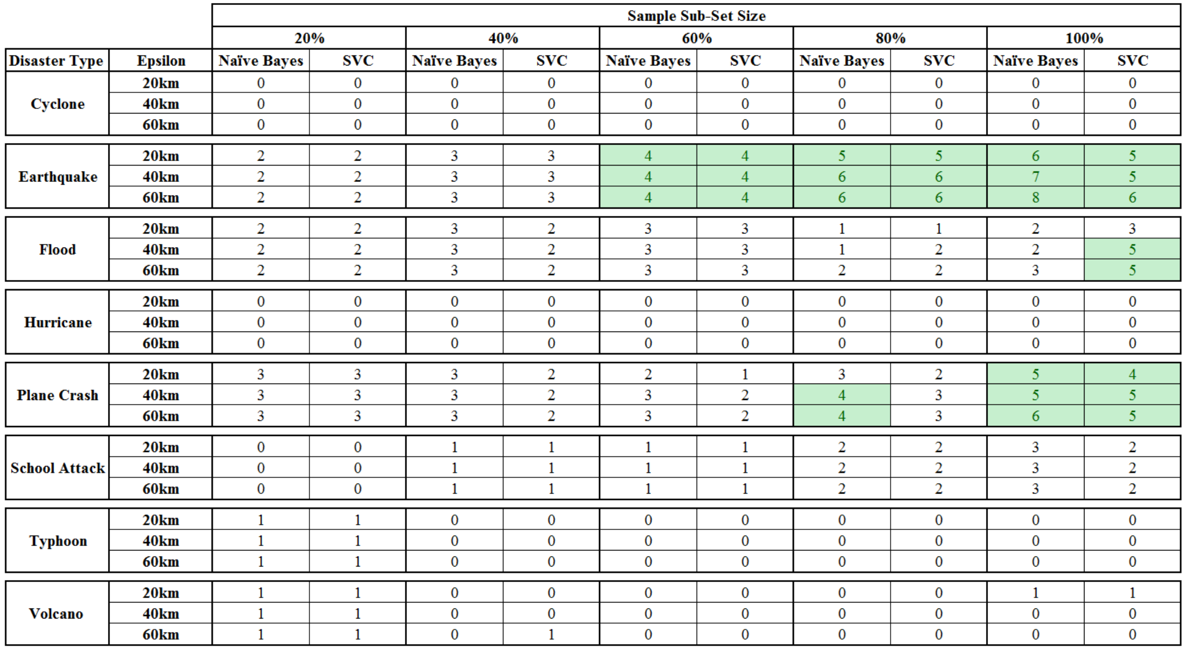
Figure 15 shows how increasing the *eps* value does not materially change the number of clusters formed, whilst keeping the *min_points* value unchanged. This is demonstrated in the maximum range in events that was detected across the three different *eps* values tested. In figure 15, any values not equal to zero (a range exists) are highlighted green:



Disaster Type	No. Points	Sample Sub-Set Size									
		20%		40%		60%		80%		100%	
		Naïve Bayes	SVC	Naïve Bayes	SVC	Naïve Bayes	SVC	Naïve Bayes	SVC	Naïve Bayes	SVC
Cyclone	20	0	0	0	0	0	0	0	0	0	0
	50	0	0	0	0	0	0	0	0	0	0
	80	0	0	0	0	0	0	0	0	0	0
Earthquake	20	0	0	0	0	0	0	1	1	2	1
	50	0	0	0	0	0	0	0	0	0	0
	80	0	0	0	0	0	0	0	0	0	0
Flood	20	0	0	0	0	0	0	1	1	1	2
	50	0	0	0	0	0	0	0	0	0	0
	80	0	0	0	0	0	0	0	0	0	0
Hurricane	20	0	0	0	0	0	0	0	0	0	0
	50	0	0	0	0	0	0	0	0	0	0
	80	0	0	0	0	0	0	0	0	0	0
Plane Crash	20	0	0	0	0	1	1	1	1	1	1
	50	0	0	0	0	0	0	0	0	1	1
	80	0	0	0	0	0	0	0	0	0	0
School Attack	20	0	0	0	0	0	0	0	0	0	0
	50	0	0	0	0	0	0	0	0	0	0
	80	0	0	0	0	0	0	0	0	0	0
Typhoon	20	0	0	0	0	0	0	0	0	0	0
	50	0	0	0	0	0	0	0	0	0	0
	80	0	0	0	0	0	0	0	0	0	0
Volcano	20	0	0	0	1	0	0	0	0	0	0
	50	0	0	0	0	0	0	0	0	0	0
	80	0	0	0	0	0	0	0	0	1	1

Fig. 15: Range in the number of disasters detected per disaster-type: vary *eps* but not *min_points* value or sample size

Figure 15 shows that the maximum range of disasters detected across disaster-types when keeping the *min_points* value constant but varying the *eps* value between 20km, 40km and 60km is two. This is for earthquakes, with a *min_points* value of 20 and an eps value of 20km, and floods, with the same *eps* and *min_points* values, both using 100% of the corpus. In both cases, the range values are obtained when the *eps* value is at 60km, which allows more tweets to be included in the region, which in turn enables the threshold *eps* value to be exceeded. Since the range of disasters detected across the majority of disaster-types is zero, it can be asserted that the *eps* value does not materially impact the performance of the *DBSCAN* algorithm.



		Sample Sub-Set Size									
		20%		40%		60%		80%		100%	
Disaster Type	Epsilon	Naïve Bayes	SVC	Naïve Bayes	SVC	Naïve Bayes	SVC	Naïve Bayes	SVC	Naïve Bayes	SVC
Cyclone	20km	0	0	0	0	0	0	0	0	0	0
	40km	0	0	0	0	0	0	0	0	0	0
	60km	0	0	0	0	0	0	0	0	0	0
Earthquake	20km	2	2	3	3	4	4	5	5	6	5
	40km	2	2	3	3	4	4	6	6	7	5
	60km	2	2	3	3	4	4	6	6	8	6
Flood	20km	2	2	3	2	3	3	1	1	2	3
	40km	2	2	3	2	3	3	1	2	2	5
	60km	2	2	3	2	3	3	2	2	3	5
Hurricane	20km	0	0	0	0	0	0	0	0	0	0
	40km	0	0	0	0	0	0	0	0	0	0
	60km	0	0	0	0	0	0	0	0	0	0
Plane Crash	20km	3	3	3	2	2	1	3	2	5	4
	40km	3	3	3	2	3	2	4	3	5	5
	60km	3	3	3	2	3	2	4	3	6	5
School Attack	20km	0	0	1	1	1	1	2	2	3	2
	40km	0	0	1	1	1	1	2	2	3	2
	60km	0	0	1	1	1	1	2	2	3	2
Typhoon	20km	1	1	0	0	0	0	0	0	0	0
	40km	1	1	0	0	0	0	0	0	0	0
	60km	1	1	0	0	0	0	0	0	0	0
Volcano	20km	1	1	0	0	0	0	0	0	1	1
	40km	1	1	0	0	0	0	0	0	0	0
	60km	1	1	0	1	0	0	0	0	0	0

Fig. 16: Range in the number of disasters detected per disaster-type: vary *min_points* but not *eps* or sample size

In figure 16 there is no material range in the number of disasters detected across disaster-types. The occasions where the value exceeds three, is largely concentrated around earthquakes and plane crashes, with a small spike for floods. This range appears to increase as the sample sub-set size increases and is most apparent at 100%. This suggests that out of the two parameters tested, the performance of the *DBSCAN* algorithm is most sensitive to *min_points*, which is directly affected by the number of tweets processed.

The location of a disaster is estimated by taking the centermost point of tweets that form a cluster (disaster). This function is only performed when the *eps* and *min_points* threshold values have been exceeded.

5.5 Location Estimation

To understand if using the centermost point of a cluster to approximate the location of a disaster is accurate, the locations estimated by the *DBSCAN* algorithm for each disaster-type are manually inspected. This is based on the .csv exports provided by the proposed tool, which outputs the location name, coordinates and disaster-type of the disaster detected. Manually inspecting the results identified that when the *min_points* value is set to a low value, such as 20, a large number of disasters are detected, which are largely spurious. An example of this is the detection of a volcano eruption in San Jose, California, USA, which is a densely populated city that was not impacted by a volcano eruption. Instead, the volcano eruption took place in Iceland. The *location estimation* of San Jose, USA was caused by a large spike in geotagged tweets that related to the volcano eruption in Iceland.

One problem faced when detecting a cluster and estimating its location is the class imbalance in the test data classified. Larger sample sub-sets led to a larger number of clusters being formed, of which many did not correspond to an actual disaster. This was particularly pronounced for earthquakes and the plane crash of flight MH370, given their class dominance in the test set used. Figure 17 shows the different estimated location names for the plane crash of flight MH370 across all *eps* values and the sample sub-set size of 20% and 100% (the smallest and largest sample sizes tested, respectively). The percentage of the sample sub-set and *min_points* are outlined, with the most accurate location underlined and bold (Chalok, Malaysia):

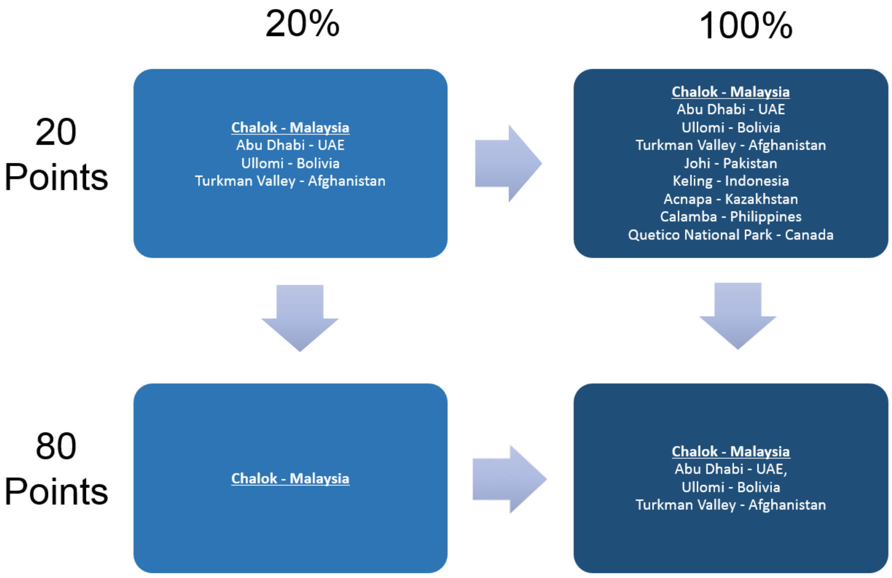


Fig. 17: Names of locations estimated by *DBSCAN* for plane crashes across all *eps* values

It can be seen that as the *min_points* threshold increases, the number of clusters detected decreases. When a 20% sample sub-set and a *min_points* value of 20 is used, four locations are detected. Once the *min_points* value is increased to 80, only one location remains, which is the most accurate estimation. When a 100% sample subset is used and a *min_points* value of 20 used, nine clusters are detected. When the *min_points* value is increased to 80, it results in the same estimations as the 20% sample sub-set with *min_points* set to 20. These outputs reinforce the need to set an appropriate *min_points* threshold, otherwise a user of the proposed tool will have to manually investigate a large number of spurious clusters.

As identified in the systematic literature review in section 2, the effectiveness of the text classifiers used directly impacts the accuracy of locations estimated by the *DBSCAN* algorithm, since any misclassified tweets contribute to the formation of spurious clusters. This is prevalent for tweets that pertain to floods, which impacted regions in India and Pakistan in 2014. Instead, the higher levels of incorrect classification by the *Naive Bayes* classifier, compared to the *SVC*, meant that in some instances, such as a 20% sample sub-set with an *eps* value of 20km and *min_points* of 20, Gandava, Pakistan is an estimated location, which actually pertains to earthquakes. It can be seen in figure 18 that a large number of locations estimated for earthquakes are also predicted for plane crashes (as can be seen figure 17) across all *eps* values tested. The duplicate locations are underlined and in bold:

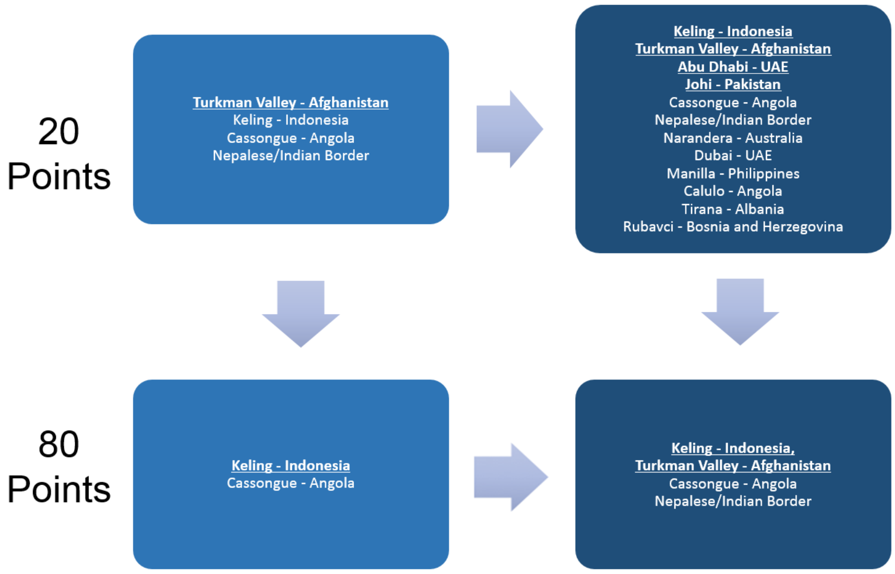


Fig. 18: Names of locations estimated by *DBSCAN* for earthquakes across all *eps* values

Since the *SVC's* superior classification performance leads to fewer clusters being formed for incorrect disaster-types, its use may reduce the number of extraneous clusters formed and improve the accuracy of the locations estimated by the *DBSCAN* algorithm.

A particularly notable observation from manually inspecting the results from the *DBSCAN* algorithm is the large volume of tweets relating to the plane crash of flight MH370, in the Malaysian city of Chalok. It is not known where flight MH370 crashed but its location has been estimated to cover an area that includes the South China Sea, the Gulf of Thailand and the Indian Ocean [52]. Across all parameter values tested for *eps*, *min_points* and sample sub-set size, Chalok, Malaysia was an estimated location. Despite this, it is suggested that this outcome provides a user with valuable information, which could be leveraged to investigate the disaster further, for what is ultimately an exceptional case, since the location of the disaster is still unknown.

Despite the aforementioned issues with the estimated locations of disasters detected using the *DBSCAN* algorithm, its overall performance is promising. Where the *min_points* value was increased to 80, across all *eps* values, a location correctly relating to a disaster-type was established for earthquakes, floods, plane crashes, school attacks and typhoons (five of the eight disaster-types in the corpus). Due to the aforementioned class imbalance, no cyclones or hurricanes were detected. More importantly, even when the full test dataset was used (a 100% sample sub-set), the maximum number of tweets pertaining to a hurricane was eight. As such, there was never enough tweets to form a cluster and detect a disaster. This lays further emphasis on the importance of choosing an appropriate *min_points* value that is tailored to the number of tweets being classified.

The heterogeneous nature of the tool's ability to detect and estimate the location of specific types of disasters is based on the dataset used for training, which contained tweets relating to eight specific disaster types. In an effort to use the tool to detect disasters not tested in the paper, the dataset could be supplemented with tweets relating to other disasters. It is important to note that this is only applicable to the text classification component of the tool. The DBSCAN algorithm used in the tool to detect and estimate the geographic location of these disasters would require no amendments, other than appropriate *eps* and *min_point* values.

5.6 Survey Results - Potential Development for the NGO Portal

In an attempt to understand the practical challenges that need to be overcome for the portal to be effective at identifying local and regional NGOs, an on-the-ground survey was conducted. The NGOs surveyed provide a spectrum of services, some of which specifically relate to disaster relief. The key themes identified were:

The portal needs to ensure that the activities of NGOs and their geographic location is as transparent and ac- curate as possible. This is to enable NGOs with appropriate expertise to be identified, since they can provide support to people adversely affected by a disaster. A major issue faced by NGOs surveyed is that the majority of their funding stems from international governments and corporations, who have specific objectives. Both of these parties have very clear objectives of what they are seeking to achieve with the funds offered. These objectives rarely align with the expertise of the NGOs seeking funds, however, since funding is scarce, NGOs may resort to adjusting their goals to satisfy the criteria specified. By ensuring that users understand the objectives of the NGOs, it can ensure that funds are being donated to NGOs who are best-positioned to provide support.

Aggregating and processing donations on the portal and sending them directly to the NGOs would help reduce the administrative burden that NGOs in India currently face when receiving international donations. Legislation implemented in India has meant that the extremely high level of documentation that must be disclosed by NGOs to receive international donations, in comparison to domestic donations, has resulted in many NGOs surveyed not seeking international donations. As such, if the portal is able to process payments on behalf of donors and aggregate them over a period of time, it could reduce the administrative burden for NGOs to receive international donations. This is recognised as a challenge, given the focus on providing local and regional NGOs with funds in the immediate aftermath of a disaster but it could prove to be advantageous if the volume of donations is large, enabling regular but sizable donations to be sent.

Using the portal to identify unscrupulous actors who are purporting to be NGOs will benefit credible NGOs in the future. By making their true identity known, the credibility of the NGO sector can be improved. This in turn can help NGOs who provide credible and reliable services to be identified and hopefully result in increased funding. This could be via an increase in direct donations but also less competition for funding from corporations - following the departure of unscrupulous NGOs from the sector - which is often the primary source of donations for NGOs in India. This stems from legislation implemented in 2014, which made it a legal requirement for Indian corporations with annual revenue over ten billion rupees (roughly equivalent to 105 million GBP), to donate 2% of their net profit to NGOs or charities.

6 Conclusion

The goal of this paper was to develop an automated *disaster detection* tool that is able to detect a disaster and accurately estimate its geographic location using information contained in tweets. Four multinomial text classifiers were tested, namely: *Logistic Regression*, *Random Forest*, linear *SVC* and *Naive Bayes*. Following 'testing', the four classifiers were reduced to two: *Naive Bayes* and a linear *SVC*. The classifiers were then analysed during 'evaluation' and the linear *SVC* with a *cost* parameter of 0.5 was selected as the most appropriate to determine the disaster-type a tweet relates to, after tweets have been normalised and converted into a unigrams *TF-IDF* vector. This decision was made in conjunction with the testing of the *DBSCAN* algorithm, since the output of the text classifiers was used as the input to the *DBSCAN* algorithm.

To holistically test the performance of the *DBSCAN* algorithm, the *eps* and *min_points* parameter values were varied. It was determined that the *eps* (radius size) was relatively insensitive and the *min_points* parameter (number of tweets) was of more importance. It is suggested that the proposed *disaster detection* tool uses *DBSCAN* parameter values of 20km for *eps* and 80 for *min_points*. The *eps* value of 20km is based on the notion of a false negative (not detecting a disaster that has occurred) being worse than a false positive (detecting a disaster that did not happen). In the case of the *min_points* parameter, the suggested value of 80 is based on the high quantity and velocity of content that is generated on Twitter. Lower values may cause the threshold to be exceeded often, causing spurious clusters to form. However, as suggested with the *eps* value of 20, the value can be adjusted if necessary.

To compliment the proposed *disaster detection* tool, a prototype portal has been built for the purpose of identifying local and regional NGOs that can assist people adversely affected by a disaster. This portal, combined with the proposed *disaster detection* tool, is considered to be a truly novel contribution, which to the authors' knowledge has not been developed already. Existing literature has shown the power of crowdsourcing and how it can be leveraged to provide powerful solutions, such as the swift identification of NGOs that can provide disaster relief to people affected by a disaster that is detected using the proposed *disaster detection* tool. A further contribution of this paper are the findings from an on-the-ground survey conducted in Uttar Pradesh, India. This survey has provided a range of development points that can be used in the future to develop the portal further.

Following the analysis of the linear *SVC* text classifier and *DBSCAN* algorithm that form the hybrid *disaster detection* tool, several areas of potential development have been identified. Despite the comprehensiveness of the corpus provided by [27], no class is given to tweets that do not pertain to a disaster. For example, a class such as 'other' would be of value, as it would prevent the selected *SVC* classifier having to assign a tweet to a disaster, if it is extraneous. Additionally, it could be argued that the *SVC* classifier should be tested on another corpus of tweets. This will help to determine whether its strong performance is a product of the methodology and techniques implemented (such as text normalisation, unigrams and *TF-IDF*), the features in the corpus of tweets (e.g. explicit language and lack of 'noise') or a combination of the two. Furthermore, the tool could further developed to process tweets in other languages, analyse more features, such as emojis, or applied to more domains, such as riots and protests.

In addition to the proposed combination of the automated *disaster detection* tool and prototype NGO identification portal, a generalised framework is proposed that categorises the different approaches that can be adopted for *disaster detection*. This tool provides a conceptual overview of how the different approaches are inter-related and the different features of each approach. It is hoped that this framework can be leveraged by researchers and practitioners in the future to develop *disaster detection* tools for the humanitarian community.

Overall, this paper has developed a powerful 'plug-and-play' *disaster detection* tool that is able to detect man-made and natural disasters across eight different disaster-types and estimate their geographic location from tweets with high precision in a scalable manner, to deal with the high volume and velocity of tweets generated on Twitter. The tool is able to overcome the issue of limited geotagged tweets by leveraging *NLP* techniques, such as *Named Entity Recognition (NER)*, to identify geographic locations, identify their coordinates and provide users of the tool with user-friendly outputs. This includes a world map with colour-coordinated plots for different disasters, alongside a .csv export, which contains the disaster-type, location name and coordinates of a disaster detected. This is all provided in a 'one-stop' manner, with fully assembled Python code that requires only a .csv file that contains raw tweets as an input.

References

1. Abel, F., Hauff, C., Houben, G.J., Stronkman, R., Tao, K.: Twitcident: fighting fire with information from social web streams. In: Proceedings of the 21st International Conference on World Wide Web, pp. 305–308. ACM (2012)
2. Achrekar, H., Gandhe, A., Lazarus, R., Yu, S.H., Liu, B.: Online social networks flu trend tracker: a novel sensory approach to predict flu trends. In: International Joint Conference on Biomedical Engineering Systems and Technologies, pp. 353–368. Springer (2012)
3. Aggarwal, C.C., Zhai, C.: A survey of text clustering algorithms. In: Mining text data, pp. 77–128. Springer (2012)
4. Ashktorab, Z., Brown, C., Nandi, M., Culotta, A.: Tweedr: Mining twitter to inform disaster response. In: ISCRAM (2014)
5. Atefeh, F., Khreich, W.: A survey of techniques for event detection in twitter. Computational Intelligence **31**(1), 132–164 (2015)
6. Baldwin, T., Li, Y.: An in-depth analysis of the effect of text normalization in social media. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 420–429 (2015)
7. Becker, H., Naaman, M., Gravano, L.: Learning similarity metrics for event identification in social media. In: Proceedings of the third ACM international conference on Web search and data mining, pp. 291–300. ACM (2010)
8. Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M., Maynard, D., Aswani, N.: Twitie: An open-source information extraction pipeline for microblog text. In: Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013, pp. 83–90 (2013)
9. Cameron, M.A., Power, R., Robinson, B., Yin, J.: Emergency situation awareness from twitter for crisis management. In: Proceedings of the 21st International Conference on World Wide Web, pp. 695–698. ACM (2012)
10. Capdevila, J., Cerquides, J., Nin, J., Torres, J.: Tweet-scan: An event discovery technique for geo-located tweets. Pattern Recognition Letters **93**, 58–68 (2017)
11. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
12. Cheng, T., Wicks, T.: Event detection using twitter: a spatio-temporal approach. PloS one **9**(6), e97807 (2014)
13. Cordeiro, M.: Twitter event detection: combining wavelet analysis and topic inference summarization. In: Doctoral symposium on informatics engineering, pp. 11–16 (2012)
14. De Albuquerque, J.P., Herfort, B., Brenning, A., Zipf, A.: A geographic approach for combining social media and authoritative data towards identifying useful information for disaster management. International Journal of Geographical Information Science **29**(4), 667–689 (2015)
15. Dhavase, N., Bagade, A.: Location identification for crime & disaster events by geoparsing twitter. In: Convergence of Technology (I2CT), 2014 International Conference for, pp. 1–3. IEEE (2014)
16. Díaz-Uriarte, R., De Andres, S.A.: Gene selection and classification of microarray data using random forest. BMC bioinformatics **7**(1), 3 (2006)
17. Dittrich, A., Lucas, C.: Is this twitter event a disaster? (2014)
18. Duan, Y., Chen, Z., Wei, F., Zhou, M., Shum, H.Y.: Twitter topic summarization by ranking tweets using social influence and content quality. Proceedings of COLING 2012 pp. 763–780 (2012)
19. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd annual meeting on association for computational linguistics, pp. 363–370. Association for Computational Linguistics (2005)
20. Goldschmidt, K.H., Kumar, S.: Reducing the cost of humanitarian operations through disaster preparation and preparedness. Annals of Operations Research pp. 1–14 (2017)
21. Guha-Sapir, D., Below, R., Hoyois, P.: Em-dat: The cred/ofda international disaster database. brussels: Université catholique de louvain (2016)
22. Habdank, M., Rodehutskors, N., Koch, R.: Relevancy assessment of tweets using supervised learning techniques: Mining emergency related tweets for automated relevancy classification. In: Information and Communication Technologies for Disaster Management (ICT-DM), 2017 4th International Conference on, pp. 1–8. IEEE (2017)
23. He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, pp. 1322–1328. IEEE (2008)

24. Hoang, T.B.N., Mothe, J.: Location extraction from tweets. Information Processing & Management **54**(2), 129–144 (2018)
25. Hsu, C.W., Chang, C.C., Lin, C.J., et al.: A practical guide to support vector classification (2003)
26. Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., Meier, P.: Extracting information nuggets from disaster-related messages in social media. In: Iscram (2013)
27. Imran, M., Mitra, P., Castillo, C.: Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Paris, France (2016)
28. Jongman, B., Wagemaker, J., Romero, B.R., de Perez, E.C.: Early flood detection for rapid humanitarian response: harnessing near real-time satellite and twitter signals. ISPRS International Journal of Geo-Information **4**(4), 2246–2266 (2015)
29. Kremer, M., van Lieshout, P., Went, R.: Doing good or doing better. Development Policies in a Globalizing World (2009)
30. Kriegel, H.P., Kröger, P., Sander, J., Zimek, A.: Density-based clustering. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **1**(3), 231–240 (2011)
31. Landwehr, P.M., Wei, W., Kowalchuck, M., Carley, K.M.: Using tweets to support disaster planning, warning and response. Safety science **90**, 33–47 (2016)
32. Lee, C.H.: Mining spatio-temporal information on microblogging streams using a density-based online clustering method. Expert Systems with Applications **39**(10), 9623–9641 (2012)
33. Lee, C.H., Wu, C.H., Chien, T.F.: Burst: a dynamic term weighting scheme for mining microblogging messages. In: International Symposium on Neural Networks, pp. 548–557. Springer (2011)
34. Li, H., Guevara, N., Herndon, N., Caragea, D., Neppalli, K., Caragea, C., Squicciarini, A.C., Tapia, A.H.: Twitter mining for disaster response: A domain adaptation approach. In: ISCRAM (2015)
35. Li, R., Lei, K.H., Khadiwala, R., Chang, K.C.C.: Tedas: A twitter-based event detection and analysis system. In: Data engineering (icde), 2012 ieee 28th international conference on, pp. 1273–1276. IEEE (2012)
36. Long, R., Wang, H., Chen, Y., Jin, O., Yu, Y.: Towards effective event detection, tracking and summarization on microblog data. In: International Conference on Web-Age Information Management, pp. 652–663. Springer (2011)
37. Maldonado, M., Alulema, D., Morocho, D., Proaño, M.: System for monitoring natural disasters using natural language processing in the social network twitter. In: Security Technology (ICCST), 2016 IEEE International Carnahan Conference on, pp. 1–6. IEEE (2016)
38. Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: Twitinfo: aggregating and visualizing microblogs for event exploration. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 227–236. ACM (2011)
39. Mathioudakis, M., Koudas, N.: Twittermonitor: trend detection over the twitter stream. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pp. 1155–1158. ACM (2010)
40. Omohundro, S.M.: Five balltree construction algorithms. International Computer Science Institute Berkeley (1989)
41. Ozdikis, O., Oğuztüzün, H., Karagoz, P.: Evidential estimation of event locations in microblogs using the dempster–shafer theory. Information Processing & Management **52**(6), 1227–1246 (2016)
42. Popescu, A.M., Pennacchiotti, M., Paranjpe, D.: Extracting events and event descriptions from twitter. In: Proceedings of the 20th international conference companion on World wide web, pp. 105–106. ACM (2011)
43. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. Journal of machine learning research **5**(Jan), 101–141 (2004)
44. Ritter, A., Clark, S., Etzioni, O., et al.: Named entity recognition in tweets: an experimental study. In: Proceedings of the conference on empirical methods in natural language processing, pp. 1524–1534. Association for Computational Linguistics (2011)
45. 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, pp. 851–860. ACM (2010)
46. Sakaki, T., Okazaki, M., Matsuo, Y.: Tweet analysis for real-time event detection and earthquake reporting system development. IEEE Transactions on Knowledge and Data Engineering **25**(4), 919–931 (2013)
47. Saleem, H.M., Al Zamal, F., Ruths, D.: Tackling the challenges of situational awareness extraction in twitter with an adaptive approach. Procedia Engineering **107**, 301–311 (2015)

48. Samant, S.S., Murthy, N.B., Malapati, A.: Bigram-based features for real-world event identification from microblogs. In: Computing, Communication and Networking Technologies (ICCCNT), 2017 8th International Conference on, pp. 1–6. IEEE (2017)
49. Sander, J., Ester, M., Kriegel, H.P., Xu, X.: Density-based clustering in spatial databases: The algorithm gdbscan and its applications. Data mining and knowledge discovery **2**(2), 169–194 (1998)
50. Sankaranarayanan, J., Samet, H., Teitler, B.E., Lieberman, M.D., Sperling, J.: Twitterstand: news in tweets. In: Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems, pp. 42–51. ACM (2009)
51. Sherchan, W., Pervin, S., Butler, C., Lai, J., Ghahremanlou, L., Han, B.: Harnessing twitter and instagram for disaster management. IBM Journal of Research and Development **61**(6), 8–1 (2017)
52. Smith, W.H., Marks, K.M.: Seafloor in the malaysia airlines flight mh370 search area. Eos, Transactions American Geophysical Union **95**(21), 173–174 (2014)
53. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information Processing & Management **45**(4), 427–437 (2009)
54. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. Journal of documentation **28**(1), 11–21 (1972)
55. Suliman, A.T., Al Kaabi, K., Wang, D., Al-Rubaie, A., Al Dhanhani, A., Ruta, D., Davies, J., Clarke, S.S.: Event identification and assertion from social media using auto-extendable knowledge base. In: Neural Networks (IJCNN), 2016 International Joint Conference on, pp. 4443–4450. IEEE (2016)
56. To, H., Agrawal, S., Kim, S.H., Shahabi, C.: On identifying disaster-related tweets: Matching-based or learning-based? In: Multimedia Big Data (BigMM), 2017 IEEE Third International Conference on, pp. 330–337. IEEE (2017)
57. Unankard, S., Li, X., Sharaf, M.A.: Emerging event detection in social networks with location sensitivity. World Wide Web **18**(5), 1393–1417 (2015)
58. Vieweg, S.E.: Situational awareness in mass emergency: A behavioral and linguistic analysis of microblogged communications. Ph.D. thesis, University of Colorado at Boulder (2012)
59. Weng, J., Lee, B.S.: Event detection in twitter. ICWSM **11**, 401–408 (2011)
60. Xu, X., Ester, M., Kriegel, H.P., Sander, J.: A distribution-based clustering algorithm for mining in large spatial databases. In: Data Engineering, 1998. Proceedings., 14th International Conference on, pp. 324–331. IEEE (1998)
61. Zhang, C., Zhou, G., Yuan, Q., Zhuang, H., Zheng, Y., Kaplan, L., Wang, S., Han, J.: Geoburst: Real-time local event detection in geo-tagged tweet streams. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 513–522. ACM (2016)
62. Zhou, X., Chen, L.: Event detection over twitter social media streams. The VLDB journal **23**(3), 381–400 (2014)