Journal of Big Data

RESEARCH Open Access

Using social media for sub-event detection during disasters

Loris Belcastro[1], Fabrizio Marozzo[1*], Domenico Talia[1], Paolo Trunfio[1], Francesco Branda[1], Themis Palpanas[2,3] and Muhammad Imran[4]

*Correspondence: fmarozzo@dimes.unical.it
[1] University of Calabria, Rende, Italy
Full list of author information is available at the end of the article

Abstract

Social media platforms have become fundamental tools for sharing information during natural disasters or catastrophic events. This paper presents SEDOM-DD (Sub-Events Detection on sOcial Media During Disasters), a new method that analyzes user posts to discover sub-events that occurred after a disaster (e.g., collapsed buildings, broken gas pipes, floods). SEDOM-DD has been evaluated with datasets of different sizes that contain real posts from social media related to different natural disasters (e.g., earthquakes, floods and hurricanes). Starting from such data, we generated synthetic datasets with different features, such as different percentages of relevant posts and/or geotagged posts. Experiments performed on both real and synthetic datasets showed that SEDOM-DD is able to identify sub-events with high accuracy. For example, with a percentage of relevant posts of 80% and geotagged posts of 15%, our method detects the sub-events and their areas with an accuracy of 85%, revealing the high accuracy and effectiveness of the proposed approach.

Keywords: Social media, Events detection, Natural disasters, Catastrophic events, Crisis computing, Disaster management, Mass emergencies, Earthquake

Introduction

Social media platforms have become an important source of information that can be exploited to understand human dynamics and behaviors. Social media posts can be geotagged, that means they are marked with geographic coordinates that allow a program to identify the location where the post was created. In some cases, such information can be combined with the textual content of the post to understand what was happening in that location. This information is extremely useful in many application contexts, such as understanding the movement of tourists within cities [1] or the behaviours of fans following important sporting events [2], discovering the best areas to open new businesses [3], analyzing the purchasing trends of users in a specific area [4].

Data elements contained in social media posts are often unstructured and require advanced analysis in order to extract useful knowledge. For example, the textual content of a post may contain information about the discussion topic [5], the sentiment of the user who wrote the posts [6], the place where the post was written [7], user opinion on a certain argument [8] and risk prevention [9]. To obtain this information, advanced

Springer Open

machine learning techniques, such as Natural Language Processing (NLP), neural networks and deep learning techniques, must be exploited [10, 11].

In the context of natural disasters, the very large use of social media platforms has enabled eyewitnesses and other disaster-affected people to share information about their damages, risks and emergencies in real time. As an example, during Hurricane Harvey in 2017, when 911, the emergency telephone number in the US, was overwhelmed by thousands of calls from those in need of immediate aid, people turned to social media to ask for help [12]. Research studies show the importance and usefulness of the information shared during disasters, both through traditional infrastructures [13] and social media [14, 15].

Despite these advantages, the use of social media posts to help rescue and intervention activities remains an open challenge as users often publish posts containing inaccurate information, slang or abbreviated words, or without using geolocalization. While extensive research work has been done on the classification of posts to understand their high-level informational categories [14], little focus has been given to understand and extract small-scale events that affect small communities. In fact, every disaster creates a series of small-scale emergencies (*sub-events*), such as family members stranded, power outage, damage to buildings, school closure, or damage to bridges. Normally, these sub-events affect only a small portion of the population in the disaster area and thus receive less attention and delayed response. Among other causes, the lack of information about these events causes a slow response from the authorities, especially during an ongoing disaster.

In this work, we aim at identifying small-scale events that occurred after a natural disaster or catastrophic event. For this purpose, we present a new method, namely SEDOM-DD (Sub-Events Detection on sOcial Media During Disasters), for detecting sub-events during disasters from social media data. Specifically, the proposed method addresses two important issues: understanding whether a post is relevant about a disaster and discovering the sub-events that occurred in the disaster area. SEDOM-DD performs these tasks in four main steps: (i) collection of posts that are potentially related to the disaster; (ii) filtering of posts to keep only the relevant ones; (iii) data enrichment by using information contained in posts to increase the number of posts for which it is possible to estimate their geolocalization; and (iv) use of clustering techniques on geotagged relevant posts for detecting sub-events.

SEDOM-DD has been evaluated with datasets of different sizes that contain real social media posts related to different natural disasters (e.g., earthquakes, floods and hurricanes). Furthermore, starting from such datasets containing real posts, we generated synthetic datasets with different features, such as different percentages of relevant posts and/or geotagged posts. Several experiments performed on both real and synthetic datasets showed that SEDOM-DD is able to identify sub-events with high accuracy both in detecting the area where they took place and in understanding the type of problem (e.g, collapsed buildings, broken gas pipes, flooding). Specifically, with a percentage of relevant posts of 80% and geotagged posts of 15% the method correctly detects the sub-events and their location areas with an accuracy of about 85%. Also in all the other configurations, our method was able to detect sub-events with high accuracy, revealing its effectiveness even dealing with noisy data.

Differently from other existing techniques, SEDOM-DD focuses on discovering sub-events that can occur as secondary effects of a disaster. For this reason, it can be integrated with existing systems for coordinating and enhancing emergency response. The detected sub-events, together with the posts and photos that made it possible to detect such events, can be analyzed and validated by a group of experts to establish the type and the priority of interventions to be carried out.

The remainder of the paper is organized as follows. Section "Related work" discusses related work. Section "Proposed method" describes the proposed method. Section "Experimental evaluation" presents the experimental evaluation of different case studies, and Sect. "Conclusions" concludes the paper.

Related work

A recent study carried out a comprehensive literature survey on the use of social media as a tool for improving damage estimation and better organizing relief operations during disasters [14]. The study also discussed the main issues in the use of social media data in disaster scenarios, such as the difficulty of processing huge amounts of data in a timely manner, the presence of unwanted or fake information, and the difficulty of collecting data describing the different stages of a disaster. Other surveys have addressed the issue of processing social media posts during mass emergencies [14, 16–18] by focusing on different aspects, such as coordinating evacuation operations [19], combining data from different sources like satellite imagery [16], and understanding how information spreads during such events [20].

Some researchers have analyzed social media traffic data for detecting earthquakes and estimating their impact area [21, 22]. For example, Avvenuti et al. [23] developed a system, namely EARS, which analyzes streaming data from Twitter for detecting seismic events. Such a system exploits a burst detection algorithm to identify earthquakes from tweets, and processes the corpus of each message for determining the impact of the seismic events on people and infrastructure. Other works focused on collecting and providing information about earthquakes currently in action. LastQuake [24] is a system that has been developed in collaboration with the European Mediterranean Seismological Centre (EMSC) that provides eyewitnesses with visual information on felt earthquakes and, at the same time, it allows to collect user feedback on the main seismic shock and its subsequent aftershocks. Sangameswar et al. [25] proposed a sentiment analysis approach for identifying the places of natural disasters (e.g., earthquakes), which could be a region, country, or continent.

An important aspect of disaster management is identifying sub-events that can take place at different locations during or after a disaster (e.g, collapsed buildings, broken gas pipes, flooding). Different studies have tried to discover sub-events from social media data using different approaches, based on both supervised and unsupervised techniques.

Some supervised techniques have been proposed for discovering sub-events after disasters. Most of them exploit weighted graph-based structures [26], TF-IDF (Term Frequency-Inverse Document Frequency) vectors [27], while others exploited neural networks for discovering, classifying and summarizing sub-events from social media data [28–30]. Supervised techniques require a manual definition of features and parameters used by the discovering algorithms. For some events, such techniques can achieve

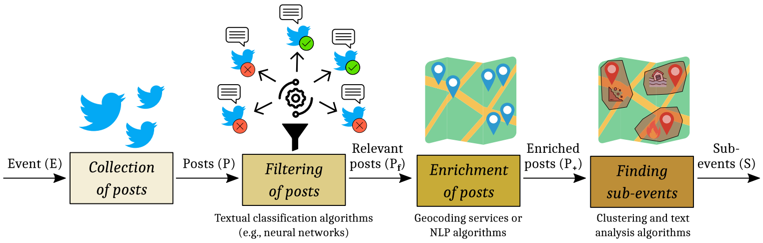


Fig. 1 Execution flow of SEDOM-DD

good results, but in many other cases the effort required to configure and optimize the algorithms could be very high and the obtained results could not be effective. For these reasons, many studies have focused on event detection techniques based on unsupervised approaches.

In fact, most unsupervised techniques that have been proposed for discovering sub-events in natural disasters are usually based on clustering algorithms coupled with similarity metrics. With regard to social media data, each textual feature (e.g., text or hashtags) is modeled as a weight vector by using TF-IDF in which the cosine similarity is used as distance metric among features [31, 32]. Other unsupervised techniques are based on topic model approaches, such as LDA (Latent Dirichlet Allocation) and HDP (Hierarchical Dirichlet Processes), which extract sub-events by analyzing the semantic representations of documents [33, 34]. Nolasco and Oliveira [35] used LDA for event mining from raw text and topic labeling methods to assign representative labels to them. Instead, Rudra et al. [36] proposed a technique based on ILP (Integer Linear Programming) and exploited a natural language processing approach for identifying and summarizing sub-events from Twitter data.

Differently from existing techniques, SEDOM-DD focuses on discovering sub-events that can occur as secondary effects of a disaster. Specifically, our method is specialized in searching and displaying sub-events on a map from social media data, even in presence of noise. The proposed method tries to use as many posts as possible by including posts that are not geotagged but that contain textual information from which geographical position can be deduced. Compared with other work that finds sub-events from social media, such as Rudra et al. [36], our method exploits a spatial clustering algorithm to identify the geographical areas where the sub-events occurred. Then, by analyzing the texts and keywords of posts in each cluster, it identifies the types of sub-events that occurred. Several experiments on different datasets related to different types of natural disasters (i.e., earthquakes, floods and hurricanes) demonstrated that SEDOM-DD is able to detect sub-events with high accuracy, revealing the effectiveness of the proposed approach.

Proposed method

To identify sub-events during a disaster, the proposed method mainly relies on four important steps. Figure 1 shows these steps together with their inputs and outputs:

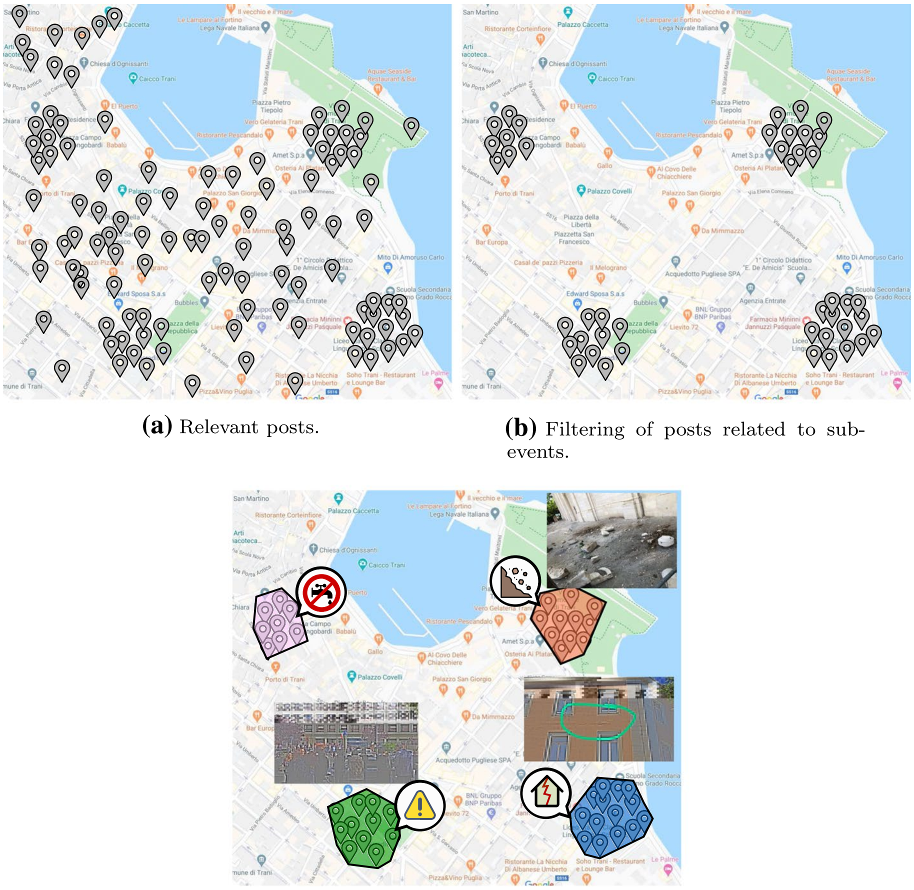
1. Data collection: given a disaster event and its impact areas, all the posts generated in the event's area are collected. These posts can be collected from social media platforms (e.g., Twitter) through queries based on keywords or locations.
2. Filtering of posts: we use supervised machine learning techniques to identify relevant posts. Posts that refer to the disaster and that come from users who live in the affected area are relevant for analysis, and thus are maintained.
3. Enrichment of posts: since many posts are relevant for analysis but are not geotagged, the information contained in the text is used to estimate the coordinates of the location where such posts were created. For example, if a post refers to a specific location (e.g., by reporting in the text the name of a road or a monument), it is possible to use a geocoding service for estimating its coordinates.
4. Finding sub-events: geotagged posts are analyzed and aggregated for finding clusters of posts mentioning a common problem (i.e., a specific sub-event that occurred in a certain area). This step involves the use of a spatial clustering algorithm to identify the areas where the sub-events occurred. Then, by analyzing the texts and keywords of posts generated inside each cluster, it is possible to understand the type of sub-event that occurred.

Figure 2 shows an example of how our method works. Specifically, it was built starting from an earthquake that really happened. On May 21, 2019, the province of Barletta-Andria-Trani in Apulia (Italy) was affected by an earthquake of magnitude 3.9 having an epicenter at 34 km of depth 4 km from the city of Barletta. After the shock warning in several municipalities across the province, many public institutions had to be evacuated, including schools, judicial offices and other facilities. The discomforts have also spread on public transport, in fact, many railway lines have been interrupted for a few hours, in order to guarantee passengers safety. The old town of the city of Trani turned out to be one of the most affected areas.

During these panic hours, we collected posts from social media focusing on the catastrophic event that occurred in the area (see Fig. 2a). Starting from such posts, those that do not explicitly refer to any sub-event have been filtered out (Fig. 2b). Posts regarding sub-events have been clustered and their text analyzed to understand the type of sub-event that occurred in each cluster's area (Fig. 2c). In particular, Fig. 2c highlights four significant sub-events that happened in the old town of Trani: the fall of material from the church of San Domenico, a structural problem with the Liceo De Santis, a water outage in the St. Mary district, and people evacuated from the judicial offices. More details on the algorithms used in the different steps of our method are provided below.

Filtering of posts

During this step, posts collected from social media are processed and filtered for keeping only the ones that are relevant for the analysis. A post is relevant when it contains text concerning a catastrophic event (e.g., earthquake, flooding) that happened in the area under analysis. Relevant posts can be further divided into two categories: (i) generic, which generically refer to a catastrophic event, without mentioning any specific sub-event (e.g., "yesterday there was an earthquake, we were very scared!"); (ii) not-generic, which explicitly refer to problems/sub-events that occurred as secondary effects



(a) Relevant posts.

(b) Filtering of posts related to sub-events.

(c) Sub-events detection.

Fig. 2 An example of using SEDOM-DD on posts collected during the earthquake in the old town of Trani (May 21, 2019)

of a catastrophic event (e.g., "we have been without electricity since yesterday"). We are mainly interested in relevant posts and, in particular, non-generic ones that mention some sub-events that have occurred.

It is evident that the classification of posts is a crucial step for obtaining accurate sub-event results. In Sect. "Experimental evaluation" we described the data we collected on Twitter and the results of some classification algorithms for separating relevant tweets from not-relevant ones. The results show that classification algorithms are able to correctly detect relevant tweets with high precision.

Enrichment of posts

The proposed method uses geotagged posts to identify the areas where the sub-events occurred. The main problem with posts from social media is that they are not always geotagged, which makes them not always useful for the analysis. The data enrichment step aims at estimating the coordinates of relevant but not geotagged posts through the analysis of the text. In this way, it is possible to increase the volume of geotagged data to be analyzed, which should lead to better accuracy in the identification of sub-events.

Posts that are not geotagged can include textual information that allows to estimate their geographical coordinates. For instance, users often report in the text the name of the street or the district where the event occurred (e.g., "Washington Street in Cork closed to traffic following the partial collapse of a building"). Several studies have proposed techniques for geotagging posts exploiting the textual information they contain [37, 38]. In addition, different geocoding services, such as Google Map[1] or Nominatim,[2] can be exploited for converting an address, even partial, into coordinates. In some cases, natural language processing techniques, e.g., based on CoreNLP,[3] can also be used for identifying the locations mentioned in the text of a post.

Our method uses the following approach for estimating the coordinates of a post. Given a geographical area to be analyzed, we exploit geocoding web services for retrieving Points-of-Interest (PoIs) in the area and the most common names used to refer to them. Then, we extract street and district names from a text through textual patterns. Once we have identified this information in the text, we translate it into coordinates with four levels of accuracy: PoI, street, district, or city. For example, a post that refers to a PoI is geotagged with the coordinates of such PoI. While a post that refers to a street (without a house number) or district is assigned to a point randomly chosen inside the street/district where the sub-event occurred. A post about a catastrophic event that cannot be associated with a specific point or area is placed at the city level.

Finding sub-events

Following the same approach proposed in [39], we used a clustering algorithm to aggregate the posts that refer to the same sub-event and discover the area where it occurred. In particular, DBSCAN [40] has been chosen for its ability to detect clusters with different sizes and shapes, tolerate noise, and be applicable on small or large data sets. Moreover, in the context of the extraction of areas or regions-of-interest, it is one of the most used algorithms in the literature [41].

For each cluster identified by DBSCAN, a procedure is carried out for identifying the sub-event that occurred in the cluster's area. In particular, we extract the keywords (and their frequency) contained in the posts from such a cluster. The keywords are then sorted by frequency. A high frequency does not necessarily denote high representative keywords, but it is a useful starting point. As an example, the keyword "earthquake" may have a higher frequency than "building collapse", although "building collapse" is evidently more representative as a sub-event that occurred in an area. The most representative keywords are then compared with a manually trained dictionary, which contains a list of terms that are commonly used to report specific sub-events that occurred. The dictionary associates a term, representing a type of sub-event, with some synonyms. As an example, for the sub-event "*collapsed house*" we also consider a list of similar terms, such as "*destroyed house*", "*house collapse*", and "*unsafe house*". As stated in [36], the terms used to report a sub-event in the text are usually composed of a pair ⟨subject entity, action happened⟩, such as "*bridge collapsed*" and "*power outage*".

[1] https://developers.google.com/maps.

[2] https://nominatim.openstreetmap.org.

[3] https://stanfordnlp.github.io/CoreNLP/.

Experimental evaluation

Several experiments were carried out to evaluate the performance of SEDOM-DD, using datasets related to different types of natural disasters (i.e., earthquakes, floods and hurricanes) that occurred in the period 2009–2019. Moreover, for evaluating the performance of SEDOM-DD using data with different characteristics and levels of precision, we started from such real data and generated a few synthetic datasets.

This section is organized as follows. Section "Collected data and classification of relevant one" describes the data collection process and the algorithms used to classify posts in *relevant* and *not relevant*. Section "Detection of sub-events on synthetic data" discusses the synthetic data generator, the algorithm for detecting sub-events, and results obtained in our tests. Finally, Sect. "Detection of sub-events on real data" presents the results obtained by SEDOM-DD on a large collection of posts about Hurricane Harvey, a Category 4 storm that hit Texas in 2017.

Collected data and classification of relevant one

In this paper, we used social media messages posted on Twitter during catastrophic events. Although our system is able to use data from other social media (e.g., Facebook or Flickr), Twitter has been chosen because it is widely used in this application context as it allows to download large amounts of data through public APIs. Other social media, although more widespread and used than Twitter (Facebook and Instagram), do not allow researchers to download users' posts on a certain topic and therefore appear to be unusable.

We used Twitter APIs for searching and collecting tweets matching keywords related to earthquakes, including those that occurred in Barletta (May 21, 2019) and Peru (May 26, 2019). From the analysis of the collected data, we noticed that some tweets report the earthquake and the problems/sub-events it generated (*relevant*), while others do not refer to the catastrophic event (*not relevant*).

Starting from the collected data, we created a manually classified dataset (D_1) composed of 5000 tweets, half *relevant* and half *not relevant*. Such data have been used to train different machine learning algorithms, which are Naïve Bayes (NB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), XGBoost (XGB), and Neural Networks (NN). In particular, we used the implementations included in the scikit-learn library[4], together with Keras[5], TensorFlow[6] and Word2Vec [42] for creating neural networks.

The obtained classification models take into account different features of tweets, such as length and presence of keywords, hashtags or bi-grams that are typically used to refer to disasters. Let $P = \{p_1, p_2, ..., p_n\}$ be a set of social media posts, where a generic post p_i is a social media content (e.g., a tweet) posted by a user after a catastrophic event E. Specifically, a generic post p_i includes:

- *user_id*, containing the identifier of the user who posted p_i;

[4] https://scikit-learn.org/stable/.

[5] https://keras.io/.

[6] https://www.tensorflow.org/.

Table 1 Hyperparameter settings

Algorithm	Hyperparameter (value)
KNN	Number of neighbors (13); type of algorithm (auto); leaf size (30); power parameter *p* (1);
SVM	C (10); Kernel (rbf); gamma (0.1);
Decision tree	Maximum depth (20); minimum samples leaf (1);
Random forest	Number of estimators (300); maximum features (auto); maximum depth (70); minimum samples split (5) minimum samples leaf (4); bootstrap (true);
XGBoost	Number of estimators (500); learning rate (0.01); maximum depth (10); minimum child weight (5);
Neural network (CNN) + Word-2vec	Batch size (64); number of epochs (100); optimizer (Adam); dropout (0.3); number of hidden layers (2); filter size (1); number of filters (200); minimum word frequency (5); iterations (100); layer size (300); window size (25);

Table 2 Datasets specifications

ID	Place	Type	Year	N. of tuples	Relevant	Not relevant
D_2	Italy	Earthquake	2019	2000	1000	1000
D_3	L'Aquila	Earthquake	2009	1062	792	270
D_4	Emilia	Earthquake	2012	3170	2648	522
D_5	Sardegna	Flood	2013	976	911	65
D_6	Genova	Flood	2014	434	388	46

- *timestamp*, indicating when (date and time) p_i was posted;
- *text*, containing a textual description of p_i;
- *tags*, containing the tags associated to p_i;
- *coordinates*, which consists of latitude and longitude of the place from where p_i was created (often this field is undefined);
- *profile_geo*, containing public location information provided by the user in its profile;
- *length*, indicating the length of the text of p_i;
- *numKeywords*, indicating the number of relevant keywords (e.g., earthquake, flooding, magnitude, lack of water, electrical problems) contained in the text of p_i;

We performed several experiments for tuning the input hyperparameters used to control the training process. Table 1 reports the values of the main hyperparameters used for the different algorithms.

For the different algorithms, the classification models have been trained using dataset $D1$. Then, such models have been tested on five datasets [43], different from $D1$, which are related to different natural disasters (i.e., floods and earthquakes) that occurred in the period 2009–2019 (see Table 2). In such a way, the training and testing datasets are completely decoupled, which enables to evaluate how well the models are generalized to deal with new unseen data. It is worth noting that some datasets are unbalanced because the two classes, *relevant* and *not relevant*, are not equally represented. In order to correctly evaluate the classification models, the training datasets have been balanced before building the models [44].

With all the datasets, the classification algorithms were able to separate relevant tweets from non-relevant ones with high accuracy. As an example, Table 3 shows

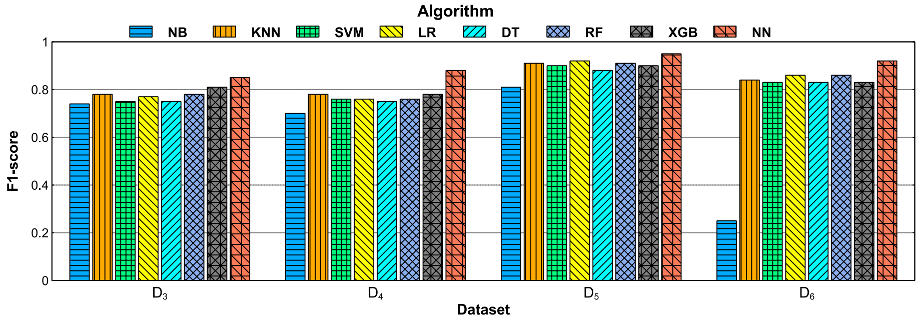


Fig. 3 Comparative analysis among several machine learning algorithms, evaluating the F1-score obtained by our approach for each dataset used in this work

Table 3 Evaluation of the classification models made on the D_2 testset

Algorithms	Acc	Prec	Rec	F1
Naïve Bayes	0.753	0.735	0.753	0.739
KNN	0.807	0.803	0.807	0.781
SVM	0.776	0.765	0.776	0.751
Logistic Regr.	0.790	0.773	0.790	0.766
Decision Tree	0.744	0.755	0.744	0.753
Random For.	0.795	0.794	0.790	0.783
XGBoost	0.815	0.812	0.815	0.809
Neural Net.	**0.830**	**0.826**	**0.864**	**0.845**

the results obtained by the different algorithms on the D_2 dataset (similar behaviors we obtain with the other datasets). The algorithm based on neural networks was the most accurate with an accuracy of 83%, followed by the algorithms XGBoost (81%) and Random Forest (80%). Figure 3 reports the classification results obtained with the other four datasets (D_3, D_4, D_5, D_6), which assess the high accuracy obtained by neural networks in all four tests. For this reason. such a model has been used for classifying posts into *relevant* and *not relevant* with high accuracy.

Detection of sub-events on synthetic data

To evaluate the performance of SEDOM-DD, we generated several synthetic datasets, each with different characteristics and levels of precision [45]. In particular, such datasets were generated starting from real social media posts published during or immediately after catastrophic events. Some of these synthetic posts are marked with precise geographic coordinates, others are not geotagged but contain information that can be used to estimate their coordinates with a varying degree of precision, and the remaining ones generically refer to the main disaster but not to any sub-events.

In the next sections we describe the algorithms used for generating synthetic data and detecting sub-events.

Table 4 Simulation parameters

Parameter	Description	Value
percGeotagged	Perc. of geotagged posts	10%, **15%**, 20%
percGeneric	Perc. of generic posts	10%, **20%**, 30%
distrGeoInfoInText	Distribution of geotagging information in the text of a post (PoI/district/street/city)	[0.1, 0.1, 0.2, 0.6] **[0.1, 0.2, 0.2, 0.5]** [0.2, 0.2, 0.2, 0.4]
numSubEvents	Number of sub-events	**5**, **10**, **20**
postPerSubEvent	Min. and max. number of posts for sub-event	[10, 50], **[20, 100]**, [30, 150]
subEventRadius	Min. and max. radius of the area where the sub-event occurred	[50, 100], **[50, 150]**, [50, 200]
distrSubEventPosts	Distribution of sub-event posts among levels (PoI/district/street/city level)	[0.6, 0.2, 0.1, 0.1] **[0.5, 0.2, 0.1, 0.1]** [0.4, 0.2, 0.1, 0.3]
numSeeds	Number of seeds to be used	**10**
numSubEvents	Number of sub-events	**5**, **10**, **20**
numSeeds	Number of seeds to be used	**10**
subEventTypes	Types of sub-events	[Damage building, ...]
analysisArea	Analysis area in which posts are generated	[Geographic coordinates ...]
dictionaries	Dictionary of terms by type	[...]

Default values are shown in bold

Synthetic post generator

Algorithm 1 shows the pseudo-code of the procedure used to generate synthetic posts. The input parameters are reported in Table 4 along with the values that were used for the experimental evaluation described in Sect. "Results". The output is composed of a set of sub-events S and a set of posts P.

At the beginning, the two sets, S and P, are initialized (line 1). A given number of sub-events (*numSubEvents*) are generated and added to S (lines 2–19). In particular, a generic sub-event s is created (line 3) and its type (s.type) is randomly chosen from *subEventTypes*, a list of predefined sub-event types (line 4). Such a list contains different types of problems/sub-events that occur after catastrophic events, such as "damaged building", "sewerage breakage", "wall collapse", "power outage", and so on.

A random point (i.e., a pair of coordinates) in the area under analysis is chosen as the center of the sub-event (line 5). Since the effects of a sub-event propagate in the surrounding area, the propagation has been modeled with four levels of precision: *Point-of-Interest (PoI)*, *street*, *district*, *city*. The level *PoI* specifies the area where the sub-event occurred (i.e., the exact area of a collapsed building) and it is represented as a circle with center in *s.coordinates* and a radius equal to *subEventRadius* (line 6). The other levels have been introduced to take into account that the effects of a sub-event propagate in the surrounding area. The area of a level contains the areas of the lower levels, that means: $PoI \subset Street \subset District \subset City$. For the sake of simplicity, starting from a sub-event at the *PoI* level, the *Street* and *district* levels are automatically generated and represented as circles with a greater radius. The area outside the district represents the *city* level. The generator establishes the number of posts to be created for a sub-event (line 7). The sub-event s is then added to S. After that, through an iterative process, the posts associated with the event *s* are generated so that they contain information with different degrees of precision (line 9–19):

- First, it is established at which level (PoI, street, district, or city) the post *p* must be geolocated (line 12). Based on this choice, and on the propagation levels defined for the sub-event, appropriate coordinates for the post are chosen (line 13). It should be noted that these coordinates are saved as *hiddenCoordinates*, because they are only used to validate the accuracy of the results, which means they are not visible to the analysis algorithm if the post is not marked as geotagged.
- Since only a certain percentage of posts is geotagged (*percGeotagged*), we randomly determine if a generated post is geolocated or not (line 14). If the post is geotagged, the *hiddenCoordinates* are saved in the *coordinates* field (line 15), which can be read by the analysis algorithm. Otherwise, the *coordinates* remain undefined (lines 16–17).
- We generate a text for each post (line 18). Specifically, such a text can include terms related to the type of sub-event, which are taken from a pre-built dictionary which contains a certain number of terms for each type of sub-event. Moreover, the text can contain information on where the sub-event happened with varying levels of accuracy (it depends on the *distrGeoInfoText* parameter). The post *p* is then added to *P* (line 19).

Eventually, a set of generic posts are generated and added to *P*, according to the parameter *percGeneric*. In such a way, it is possible to add some noise into the data to be analyzed (line 20).

ALGORITHM 1: Synthetic post generator.

```
   Input  : Parameters present in Table 4
   Output: List of <subEvents> $S$, List of <post> $P$
 1 $S \leftarrow \emptyset$ $P \leftarrow \emptyset$
 2 for $i \leftarrow 0$ to $numSubEvents$ by $1$ do
 3     $s \leftarrow$ CreateSubEvent()
 4     $s$.type = SelectARandomType($subEventTypes$)
 5     $s$.coordinates = GetRandomCoordinates($analysisArea$)
 6     $s$.radius = GetRandomRadius($subEventRadius$)
 7     $s$.numPosts = GetRandomNumber($postPerSubEvent$)
 8     $S$.add($s$)
 9     for $i \leftarrow 0$ to $s.numPosts$ by $1$ do
10         $p \leftarrow$ CreatePost()
11         $p.subEvent$ = s
12         $p$.precisionLevel = GetRandomLevel($distrSubEventPosts$)
13         $p$.hiddenCoordinates = GetRandomCoordinates($s.coordinates$, $s.radius$,
            $p.precisionLevel$)
14         if GetRandom()<=$percGeotagged$ then
15             $p$.coordinates = $p$.hiddenCoordinates
16         else
17             $p$.coordinates = N.D.
18         $p$.text = GetRandomText(GetDictionary($dictionaries$, $s.type$),
            $distrGeoInfoText$)
19         $P$.add($p$)
20 $P$.addAll($percGeneric$, GetDictionary($dictionaries$, $generic$))
21 return $S$, $P$
```

Sub-event detection

Algorithm 2 shows the pseudo-code of the procedure used to discover sub-events from social media posts. The input is a list of posts P and the parameters of a clustering algorithm. In particular, DBSCAN was chosen as a clustering algorithm since it is resistant to noise and it can find clusters of different sizes and shapes. DBSCAN requires the following parameters as input: *eps*, the radius of the neighborhood of a point; and *minPts*, the minimum number of points that are required to form a cluster. The output is a list of sub-events S_{found} that have been discovered in the area under analysis. Regarding the resources required to run DBSCAN instances, its computational complexity is $O(n^2)$ where n is the number of points, which drops to $O(n \log n)$ if a spatial index is used [46].

We point out that, in order to obtain a real situation, not all generated posts are geotagged: only a small part of them include a geographic position or contain textual information that allows to estimate, with a certain precision, where the sub-event occurred. Therefore, due to the way the synthetic datasets have been generated, it is reasonable to expect that the sub-event detection algorithm will never reach 100% accuracy as some data is missing and cannot be reconstructed.

The algorithm analyzes the posts P by performing some preprocessing and data enrichment operations (lines 1–11). First, both not relevant and generic posts are filtered out (lines 2–3). This means they are not considered during the clustering phase. Then, all posts that are not geotagged are processed in an attempt to estimate their coordinates based on the textual information they contain (lines 4–11). According to a certain distribution, the geolocation can be estimated at the PoI level, which allows the estimation of the post coordinates with the highest precision, or at the street/district levels. Otherwise, the posts that cannot be geolocated are discarded (lines 10–11). At the end of this process, the remaining posts are thus relevant and geotagged.

In the second part of the algorithm, geotagged posts are transformed into coordinates and analyzed by DBSCAN so as to generate a set of clusters CP (lines 12–13). For each cluster $cp \in CP$, the following operations are carried out (lines 14–19). The most frequent words in the texts of posts belonging to cp are extracted (line 15). From such words, the most representative ones are selected by using the TF-IDF algorithm [47] (line 16) and compared with a dictionary containing information that allows to identify the type of event that occurred (line 17). The points included in cp can be converted into a convex polygon, which represents the area where the sub-event occurred (line 18). The detected sub-event s is added to S_{found} (line 19).

To evaluate the accuracy of the sub-event detection algorithm, we compare the sub-events found by Algorithm 2 (S_{found}) with those generated by Algorithm 1 (S). In particular, each sub-event found is compared with the one in the initial dataset that provides the largest match. Then, some performance metrics (i.e., precision, recall, and F1-score) are measured by calculating the posts that have been successfully classified as part of the sub-event.

ALGORITHM 2: Sub-event detection.

```
Input  : List of <post> $P$, $eps$ and $minNumPoints$
Output: List of <subEvents> $S_{found}$,
foreach $p \in P$ do
    if IsNotRelevant($p$) or IsGeneric($p$) then
        $P$.delete($p$)
    if $p.coordinates == N.D.$ then
        $geoInfo = p$.text.GeoInfo()
        if $geoInfo == PoI$ then
            $p$.coordinates = PoI.coordinates
        else if $geoInfo == street$ or $geoInfo == district$ then
            $p$.coordinates = GetRandom (street.coordinates or district.coordinates)
        else
            $P$.delete($p$)
$C \leftarrow$ GetCoordinates($P$)
$CP \leftarrow$ DBSCAN($C$, $eps$, $minNumPoints$)
foreach $cp \in CP$ do
    $K \leftarrow$ getkeywords($cp$)
    $K_m \leftarrow$ TF-IDF($K$)
    $s.problem \leftarrow$ subvent($K_m$, subEventDictionary)
    $s.area \leftarrow$ CONVEXHULL($CP$)
    $S_{found} \leftarrow S_{found} \cup s$
return $S_{found}$
```

Example of generated/processed data

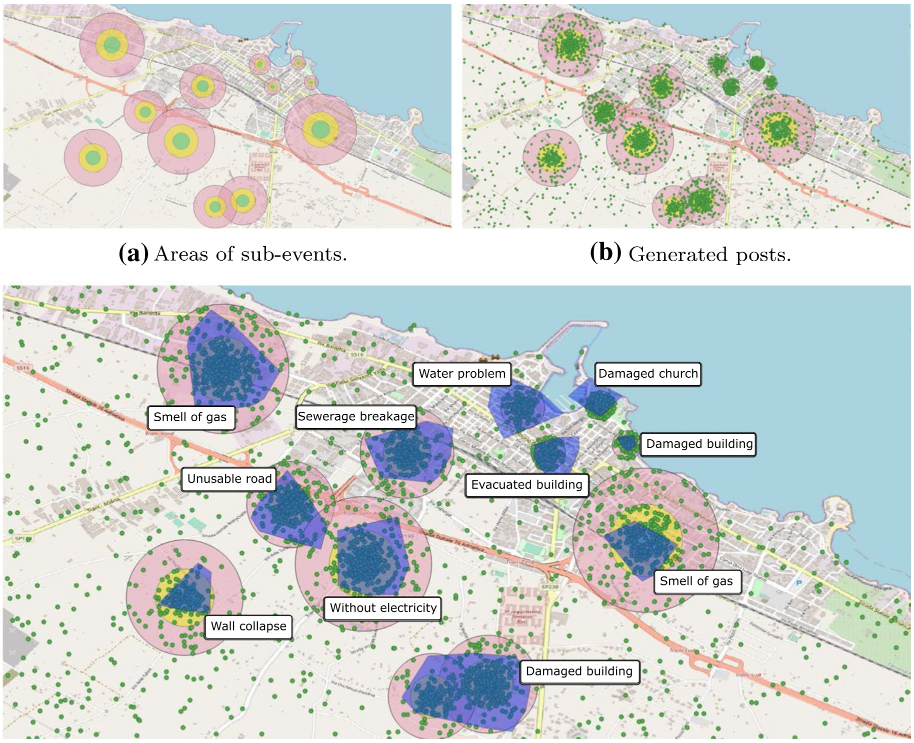
Figure 4 shows an example of synthetic data generated in the city of Trani (Apulia, Italy). Figure 4a shows some sub-events that have been represented using the four-levels model described above. Specifically, the level *PoI* is depicted in green, the *street* in yellow, the *district* in pink, and outside the pink circle we have the level *city*. Figure 4b illustrates synthetic posts (green dots) that have been generated for the different sub-events in the area. Five examples of posts have been reported in Table 5: one is geotagged (tweet ID 1), two contain texts that allow to estimate geotagging information (tweet IDs 2-3), while others are generic and do not allow to deduce the geographical coordinates (tweet IDs 4-5).

By applying DBSCAN on collected posts, it is possible to discover clusters that represent the geographical areas where the sub-events occurred. Then, a textual analysis of the posts of each cluster permits to find the main keywords used in that area so as to understand the sub-problem that has occurred. As shown in Fig. 4c, each cluster that is found is represented as a purple polygon. A label describing the occurred sub-event is also assigned to each cluster.

Results

The evaluation was carried out on synthetic datasets by using different configuration values for the parameters reported in Table 4, some of which were extracted from real Twitter data as described in [48].

Since such datasets are characterized by significant variability in the density of points (number of posts per area unit), we made several preliminary tests to determine the optimal input parameters of DBSCAN. In particular, the maximum distance between points



(a) Areas of sub-events.

(b) Generated posts.

(c) Detection of sub-events (i.e., polygons in purple) and extraction of keywords. To each cluster is assigned a label that describe the occurred sub-event.

Fig. 4 An example of post generation and analysis through SEDOM-DD

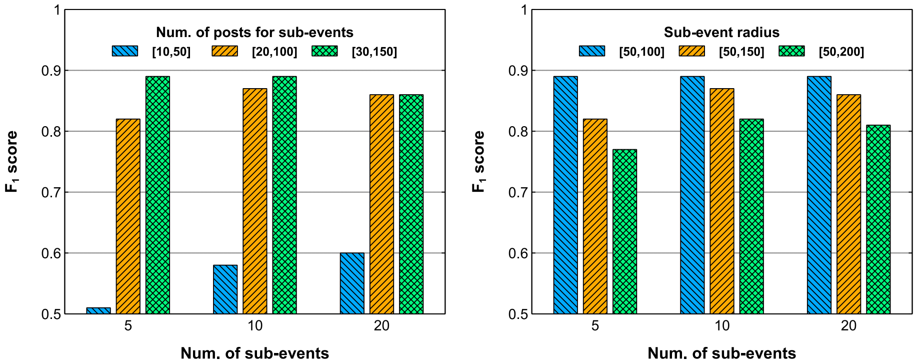
Table 5 Example of generated tweets

ID	Tweet	Lat./lon.	Est. lat./lon.
1	The building where I live was damaged	***41.268683/ 16.419321***	–
2	Evacuated from our apartment on ***Corso Vittorio Emanuele***! #earthquake #Trani	–	41.276308/ 16.417803
3	Sewage smell in the area of ***Nassiriya square***	–	41.278140/ 16.413285
4	Some buildings have been damaged during the #earthquake	–	–
5	The earthquake tonight was truly severe	–	–

Tweet#1 is geotagged, tweets #2 and #3 contain information in the text that can be used to estimate coordinates (highlighted in bold), the others cannot be geolocated

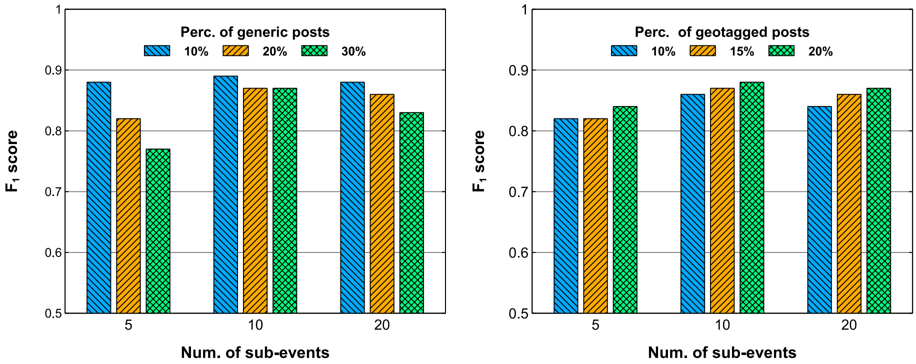
(*eps*) has been set to 7 meters and the minimum number of cluster points (*minPts*) to 150.

During our experiments, we used a reference configuration C_{ref} that has been made up with the parameter values shown in bold in Table 4 (e.g., $percGeotagged = 15\%$, $percGeneric = 20\%$). Subsequently, some parameters of such configuration have been varied to understand the behavior of our method with data more or less precise. For each parameter configuration, we performed ten tests by using different seeds.



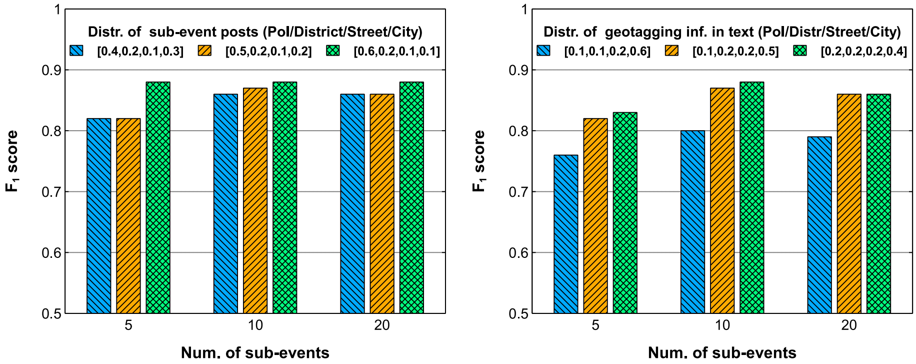
(a) $postPerSubEvent$ (b) $poiRadius$

Fig. 5 F1-score for different values of the number of posts per sub-event (*postPerSubEvent*) and radius in meters (*poiRadius*) of a sub-event



(a) $percGeneric$ (b) $percGeotagged$

Fig. 6 F1-score for different values of the percentages of generic posts (*percGeneric*) and geotagged posts (*percGeotagged*) in the data



(a) $distrSubEventPosts$ (b) $distrGeoInfoInText$

Fig. 7 F1-score for different percentage distributions among the four levels (*PoI*, *street*, *district*, *city*) of geotagged posts (*distrSubEventPosts*) and textual information that allows to estimate coordinates of a post (*distrGeoInfoInText*)

Figures 5, 6 and 7 show the behavior of SEDOM-DD in detecting 5, 10 and 20 sub-events by varying a parameter of the reference configuration C_{ref} (e.g., percentage of geotagged posts). Using the standard configuration C_{ref}, SEDOM-DD obtained an F1-score of 0.82 with 5 sub-events, 0.88 with 10 sub-events, and 0.86 with 20 sub-events (see the orange bar in the figures).

Figure 5a shows the F1-score obtained by varying the number of posts generated for each sub-event. As shown, the F1-score grows up as the number of posts for each event increases. Considering the configuration with 10 events, we obtained an F1-score of 0.58 by using the configuration with the minimum and maximum number of posts for sub-event $postPerSubEvent = [10, 50]$ (blue bar), 0.87 for [20, 100] (orange bar), and 0.89 for [30, 150] (green bar). The greater precision is due to the fact that there are more posts for each sub-event and therefore the cluster is more precise.

Figure 5b shows per F1-score obtained by varying the value of the parameter *poiRadius*. By increasing the mean radius of an event, the F1-score tends to decrease because the recall tends to be smaller (i.e., the clusters tend to be smaller). As the radius increases, the points are distributed over a larger area. As a result, there is a reduction in the density of points that could reduce the ability of DBSCAN to find larger clusters.

Other experiments have been performed to study how the results vary with the percentage of generic (*percGeneric*) and geotagged (*percGeotagged*) posts.

In particular, as the percentage of generic posts increases the F1-score decreases since there are fewer points that can be processed by DBSCAN (Fig. 6a). Instead, as the percentage of geotagged posts increases, the overall F1-score improves since DBSCAN can exploit a higher number of points to find more accurate clusters (Fig. 6b). It should be also noticed that the performance decreases as the number of sub-events in the area increases. This behavior is mainly due to the fact that the presence of multiple events in the area produces clusters that are overlapped and, for this reason, not so representative for the single sub-event.

Figure 7a shows the results obtained by changing the distribution of the geotagged posts among the four levels (PoI/district/street/city). As shown, increasing the concentration of points at the *PoI* level leads to a better F1-score since the clustering algorithm is able to work on more defined clusters (see green bar). Instead, Fig. 7b presents the results obtained by varying the percentage distribution of textual information, which allows to estimate the coordinates of a post within one of the four levels (PoI/district/street/city). Also in this case, the greater the accuracy of the information in the text (i.e., more points can be estimated at the level PoI) the greater the ability of the algorithm to discover accurate clusters for sub-events. However, increasing the number of sub-events in the area could result in a reduction of the F1-score due to the overlap of different clusters (see the case with 20 events).

The good results obtained by SEDOM-DD are most likely due to the data enrichment process, which allows DBSCAN to identify clusters with greater accuracy. In fact, as shown in Fig. 8, without using the data enrichment procedure, the DBSCAN uses only a few natively geotagged posts, obtaining a very low F1-score (e.g., 0.21 with 10 sub-events).

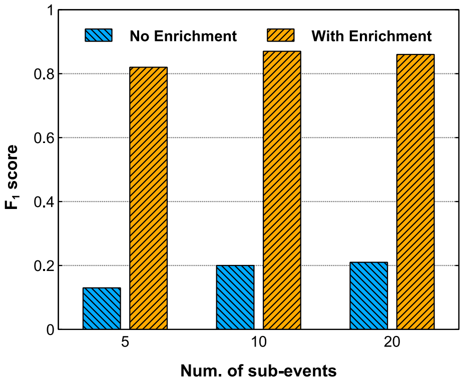


Fig. 8 Comparison of the F1-score obtained by SEDOM-DD with and without the enrichment of posts

Detection of sub-events on real data

For assessing the usability of our method in other real study cases, we carried out an analysis of a large dataset containing tweets about Hurricane Harvey, a Category 4 storm that hit Texas in 2017, causing about USD 200 billion in damage, and at least 82 deaths according to the Texas Department of Public Safety. Such a dataset contains about 6.7 million of tweets, which have been collected from August 25, 2017 to September 5, 2017, using specific keywords (i.e., "*Hurricane Harvey*", "*Harvey*", "*HurricaneHarvey*") as described in [49].

The classification model discussed in Sect. "Collected data and classification of relevant one" has been used for *filtering posts* so as to separate the relevant tweets from the not relevant ones. In particular, we classified 1,905,585 tweets as relevant (i.e., 29% of total data), but just a small part of them are geolocated (less than 1%). Through the *post enrichment* phase, we were able to deduce the location where the post has been created by analyzing the text of the tweet (15% of total data). Specifically, we used a name entity recognition algorithm based on CoreNLP for identifying the locations mentioned in the text of the tweets. A clustering algorithm is used for detecting interesting clusters on such geotagged posts. Then, a procedure is carried out for identifying sub-events by analyzing the keywords (and their frequency) contained in the posts that fall into each cluster.

Since the area under analysis is very large (several American cities were hit by the hurricane) and the posts do not provide detailed geolocation information, the clusters obtained coincide with the main cities of Texas. Considering all the clusters, the posts that report sub-events are approximately 113,346 (approximately 2 % of the total data), mainly reporting damages to infrastructures (e.g., roads or houses) or to utility services (e.g., power outages or water pipes).

Figure 9 shows some significant sub-events that were discovered. In particular, two large areas with a high density of sub-events (red areas) have been discovered in the cities of Houston and Rockport. Other areas, smaller and with a lower density of sub-events (blue-green areas), have been identified elsewhere. Table 6 reports the sub-events that have been identified in the main cities involved in the disaster. Notably, Houston was found to be the city with the highest number of sub-events that occurred after the passing of Harvey, including flooded houses and damages to toxic

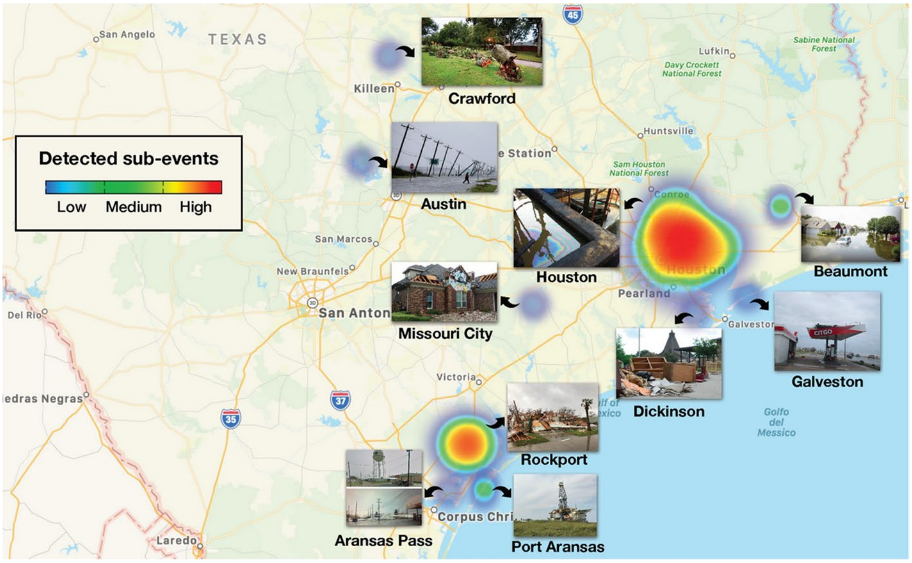


Fig. 9 Sub-events detected by SEDOM-DD using tweets collected after Hurricane Harvey

Table 6 Main sub-events detected in tweets about Harvey

City	Types of sub-events
Houston	Flooded houses, airports runways and highways, damaged toxic waste sites and electrical station, destroyed cars
Rockport	Damaged boat storage, collapsed houses, power line down
Beaumont	Flooded houses, damaged oil refineries
Port Aransas	Collapsed houses, damaged ferries and vehicles, power line down
Austin	Power outage
Crawford	Downed trees, collapsed houses
Dickinson	Flooded houses and roads, destroyed churches
Missouri City	Roofless houses, big trees down
Aransas Pass	Water service down
Galveston	Damaged gas station

waste sites. Also Rockport reported a high number of sub-events, such as collapsed houses, power lines downs, and damaged boats. The obtained results confirm that SEDOM-DD is able to discover a high number of sub-events that occurred after a large-scale natural disaster.

Conclusions

The widespread use of social media allows people who are victims of disasters (e.g., earthquakes, fires) to share real time information about damages, problems, and sub-events that can take place at different locations after a disaster (e.g, collapsed buildings, broken gas pipes). This valuable information is known only to people located where the events occurred and can be shared with rescue teams and authorities that are far away from the area. In this paper we presented SEDOM-DD, a new method that combines text mining and clustering analysis for discovering critical sub-events from social media data during natural disasters.

Several experiments have been carried out on both real and synthetic datasets for evaluating the performance of SEDOM-DD. In particular, an analysis of a large dataset containing real tweets about Hurricane Harvey showed that SEDOM-DD was able to discover a large number of sub-events that occurred after the disaster. Moreover, other experiments on synthetic datasets demonstrated that SEDOM-DD is able to identify sub-events with a very good F1-score (greater than 85%), which confirms the high accuracy and effectiveness of the proposed approach.

For this reason, SEDOM-DD can be integrated with existing systems for coordinating and enhancing emergency response. The detected sub-events, together with the posts and photos that made it possible to detect such events, can be analyzed and validated by a group of experts to establish the type and the priority of interventions to be carried out.

Acknowledgements
This work has been partially supported by the ASPIDE Project funded by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 801091.

Authors' information
L. Belcastro is a research fellow of computer engineering at the University of Calabria. F. Marozzo is an assistant professor of computer engineering at the University of Calabria. D. Talia is a professor of computer engineering at the University of Calabria and an adjunct professor at Fuzhou University. P. Trunfio is an associate professor of computer engineering at the University of Calabria. F. Branda is PhD student of computer engineering at the University of Calabria. T. Palpanas is a Senior Member of the French University Institute (IUF). M. Imran is a Research Scientist at the Qatar Computing Research Institute (QCRI).

Authors' contributions
All the authors contributed to the structuring of this paper. LB and FM designed and implemented the method and performed experimental evaluations. DT and PT led the research activities and reviewed the content of the paper. FB contributed heavily to the review of the paper by dealing with experimental evaluations. TP and MI set up and reviewed the content of the paper. All authors read and approved the final manuscript.

Availability of data and materials
The data that support the findings of this study are publicly available, since they can be gathered through Twitter APIs available at https://developer.twitter.com/. For the purpose of using the code of our method, an open-source prototype of SEDOM-DD is available at https://github.com/SCAlabUnical/SEDOM-DD.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]University of Calabria, Rende, Italy. [2]Université de Paris, Paris, France. [3]French University Institute (IUF), Paris, France. [4]Qatar Computing Research Institute, Ar-Rayyan, Qatar.

Received: 23 December 2020 Accepted: 15 May 2021
Published online: 02 June 2021

References

1. Belcastro L, Marozzo F, Talia D, Trunfio P. Parsoda: High-level parallel programming for social data mining. Soc Netw Anal Min. 2018;9(1):1–19.
2. Cesario E, Marozzo F, Talia D, Trunfio P. Sma4td: a social media analysis methodology for trajectory discovery in large-scale events. Online Soc Netw Media. 2017;3–4:49–62.

3. Ancillai C, Terho H, Cardinali S, Pascucci F. Advancing social media driven sales research: establishing conceptual foundations for b-to-b social selling. Indus Market Manage. 2019;82:293–308.
4. Shen C-w, Chen M, Wang C-c. Analyzing the trend of o2o commerce by bilingual text mining on social media. Comput Human Behav. 2019;101:474–83. https://doi.org/10.1016/j.chb.2018.09.031.
5. Athira B, Jones J, Idicula SM, Kulanthaivel A, Zhang E. Annotating and detecting topics in social media forum and modelling the annotation to derive directions—a case study. J Big Data. 2021;8(1):1–23.
6. Sarlan A, Nadam C, Basri S. Twitter sentiment analysis. In: Proceedings of the 6th IEEE international conference on information technology and multimedia; 2014. p. 212–6.
7. Middleton SE, Kordopatis-Zilos G, Papadopoulos S, Kompatsiaris Y. Location extraction from social media: geoparsing, location disambiguation, and geotagging. ACM Trans Inform Syst (TOIS). 2018;36(4):1–27.
8. Belcastro L, Cantini R, Marozzo F, Talia D, Trunfio P. Learning political polarization on social media using neural networks. IEEE Access. 2020;8(1):47177–87.
9. Subroto A, Apriyana A. Cyber risk prediction through social media big data analytics and statistical machine learning. J Big Data. 2019;6(1):1–19.
10. Roccetti M, Delnevo G, Casini L, Mirri S. An alternative approach to dimension reduction for pareto distributed data: a case study. J Big Data. 2021;8(1):1–23.
11. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. J Big Data. 2021;8(1):1–74.
12. Villegas C, Martinez M, Krause M. Lessons from Harvey: crisis informatics for urban resilience. Rice University Kinder Institute for Urban Research; 2018. p. 1–20.
13. Raza M, Awais M, Ali K, Aslam N, Paranthaman VV, Imran M, Ali F. Establishing effective communications in disaster affected areas and artificial intelligence based detection using social media platform. Fut Gen Comput Syst. 2020;112:1057–69.
14. Nazer TH, Xue G, Ji Y, Liu H. Intelligent disaster response via social media analysis a survey. ACM SIGKDD Explor Newsl. 2017;19(1):46–59.
15. Simon T, Goldberg A, Adini B. Socializing in emergencies-a review of the use of social media in emergency situations. Int J Inform Manage. 2015;35(5):609–19.
16. Said N, Ahmad K, Riegler M, Pogorelov K, Hassan L, Ahmad N, Conci N. Natural disasters detection in social media and satellite imagery: a survey. Multimed Tools Appl. 2019;78(22):31267–302.
17. Imran M, Castillo C, Diaz F, Vieweg S. Processing social media messages in mass emergency: a survey. ACM Comput Surveys (CSUR). 2015;47(4):1–38.
18. Wang Z, Ye X. Social media analytics for natural disaster management. Int J Geogr Inform Sci. 2018;32(1):49–72.
19. Slamet C, Rahman A, Sutedi A, Darmalaksana W, Ramdhani MA, Maylawati DS. Social media-based identifier for natural disaster. IOP Conf Ser Mater Sci Eng. 2018;288:012039.
20. Dong R, Li L, Zhang Q, Cai G. Information diffusion on social media during natural disasters. IEEE Trans Comput Soc Syst. 2018;5(1):265–76.
21. Crooks A, Croitoru A, Stefanidis A, Radzikowski J. Earthquake: Twitter as a distributed sensor system. Trans GIS. 2013;17(1):124–47.
22. 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, 2010; p. 851–860
23. Avvenuti M, Cresci S, Marchetti A, Meletti C, Tesconi M. Ears (earthquake alert and report system) a real time decision support system for earthquake crisis management. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014; p. 1749–1758
24. From rapid information to global seismic risk reduction. Lastquake. Int J Disaster Risk Reduction. 2018;28:32–42.
25. Sangameswar M, Rao MN, Satyanarayana S. An algorithm for identification of natural disaster affected area. J Big Data. 2017;4(1):1–11.
26. Meladianos P, Xypolopoulos C, Nikolentzos G, Vazirgiannis M. An optimization approach for sub-event detection and summarization in twitter. In: European Conference on Information Retrieval, 2018; p. 481–493. Springer.
27. Abhik D, Toshniwal D. Sub-event detection during natural hazards using features of social media data. In: Proceedings of the 22nd International Conference on World Wide Web. WWW'13 Companion, pp. 783–788. Association for Computing Machinery,New York, NY, USA 2013.
28. Nguyen DT, Al Mannai KA, Joty S, Sajjad H, Imran M, Mitra P. Robust classification of crisis-related data on social networks using convolutional neural networks. In: Eleventh International AAAI Conference on Web and Social Media 2017.
29. Wang Z, Zhang Y. A neural model for joint event detection and summarization. In: IJCAI, 2017; p. 4158–4164.
30. Bekoulis G, Deleu J, Demeester T, Develder C. Sub-event detection from twitter streams as a sequence labeling problem. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 745–750. Association for Computational Linguistics, Minneapolis, Minnesota 2019.
31. Pohl D, Bouchachia A, Hellwagner H. Automatic sub-event detection in emergency management using social media. In: Proceedings of the 21st International Conference on World Wide Web, 2012; p. 683–686 .
32. Abhik D, Toshniwal D. Sub-event detection during natural hazards using features of social media data. In: Proceedings of the 22nd International Conference on World Wide Web, 2013; p. 783–788 .
33. Xing C, Wang Y, Liu, J, Huang Y, Ma W-Y. Hashtag-based sub-event discovery using mutually generative lda in twitter. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI'16, pp. 2666–2672. AAAI Press, Phoenix, Arizona, USA 2016.
34. Srijith P, Hepple M, Bontcheva K, Preotiuc-Pietro D. Sub-story detection in twitter with hierarchical dirichlet processes. Inform Process Manage. 2017;53(4):989–1003.
35. Nolasco D, Oliveira J. Subevents detection through topic modeling in social media posts. Fut Gen Comput Syst. 2019;93:290–303.

36. Rudra K, Goyal P, Ganguly N, Mitra P, Imran M. Identifying sub-events and summarizing disaster-related information from microblogs. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. SIGIR '18. New York: NY, USA; 2018. p. 265–74.
37. Lourentzou I, Morales A, Zhai C. Text-based geolocation prediction of social media users with neural networks. In: 2017 IEEE International Conference on Big Data (Big Data), 2017; p.696–705 IEEE.
38. Zhang W, Gelernter J. Geocoding location expressions in twitter messages: a preference learning method. J Spatial Inform Sci. 2014;2014(9):37–70.
39. Belcastro L, Kechadi MT, Marozzo F, Pastore L, Talia D, Trunfio P. Parallel extraction of regions-of-interest from social media data. Concurr Comput Pract Exp. 2021;33(8);e5638.
40. Ester M, Kriegel H-P, Sander J, Xu X, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. KDD. 1996;96:226–31.
41. Belcastro L, Marozzo F, Talia D, Trunfio P. G-RoI: automatic region-of-interest detection driven by geotagged social media data. ACM Trans Knowl Discov Data. 2018;12(3):27–12722.
42. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. arXiv:1310.4546 (2013)
43. Cresci S, Tesconi M, Cimino A, Dell'Orletta F. A linguistically-driven approach to cross-event damage assessment of natural disasters from social media messages. In: Proceedings of the 24th International Conference on World Wide Web, 2015; p. 1195–1200.
44. Kotsiantis S, Kanellopoulos D, Pintelas P, et al. Handling imbalanced datasets: a review. GESTS Int Trans Comput Sci Eng. 2006;30(1):25–36.
45. Cooper C, Zito M. Realistic synthetic data for testing association rule mining algorithms for market basket databases. In: Kok JN, Koronacki J, Lopez de Mantaras R, Matwin S, Mladenič D, Skowron A, editors. Knowledge discovery in databases: PKDD 2007, 2007; p. 398–405 .
46. Gan J, Tao Y. On the hardness and approximation of Euclidean dbscan. ACM Trans Database Syst (TODS). 2017;42(3):1–45.
47. Ramos J., et al. Using tf-idf to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning, 2003; 242, p. 133–142 . New Jersey, USA.
48. Huang B, Carley KM. A large-scale empirical study of geotagging behavior on twitter. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2019; p. 365–373.
49. Alam F, Ofli F, Imran M, Aupetit M. A twitter tale of three hurricanes: Harvey, irma, and maria. Rochester, USA: Proc. of ISCRAM; 2018.

Publisher's Note