SMDRM: A Platform to Analyze Social Media for Disaster Risk Management in Near Real Time

Valerio Lorini,[1,2] **Emanuele Panizio,** [3] **Carlos Castillo** [2,4]
[1] European Commission - Joint Research Centre, Ispra. IT
[2] Universitat Pompeu Fabra, Barcelona, ES
[3] Arcadia SIT S.r.l. , Vigevano, IT
[4] ICREA, Barcelona, ES
valerio.lorini@ec.europa.eu, emanuele.panizio@ext.ec.europa.eu, chato@icrea.cat

Abstract

Social media has been described as a mechanism for understanding a situation using information spread across many minds, i.e., a form of distributed cognition (Hutchins 1995). Gaining situational awareness in a disaster is critical and time-sensitive. Social media provides a vast data source that might help improve response in the early hours and days of a crisis. However, social media platforms may not provide information in a manner that is useful for crisis responders. SMDRM is a software platform that streamlines the processing of text and images extracted from Twitter in near real-time during a specific event.

1 Introduction

The interactions among people in social media are a form of collective intelligence, as they allow us to make sense of a developing event collectively (Hutchins 1995). In the immediate aftermath of a crisis, finding relevant information is of utmost importance, Particularly in the first 12-24 hours, when often neither authoritative data nor Earth Observation (EO) images are available. For this reason, we developed a scalable, multi-modal, and multilingual platform to streamline the automated processing of messages and images in near real-time. We named it the Social Media for Disaster Risk Management (SMDRM) platform

The data are **collected** using keywords and locations based on daily forecasts from the early warnings systems or triggered manually in case of earthquakes or not-forecasted events. Then, the text is automatically **annotated** with multilingual classifiers trained in 12 languages and extended with multilingual embeddings. Simultaneously, a multi-class convolutional neural network labels relevant images for floods, storms, earthquakes, and fires (Rufolo, Muraro, and Lorini 2021). Finally, messages are **geocoded** with a two-step algorithm; location candidates are selected using a multilingual named-entity recognition tool and then searched on available gazetteers. After the platform processing, relevant information can be aggregated in spatial (administrative areas) and temporal (daily) units.

SMDRM could offer timely, valuable information to reduce uncertainty and provide added-value information such as reports or descriptions of the situation on the ground. Various stakeholders may access data to complement those extracted from traditional sensors or EO. The platform can adapt to cope with surges in workload as it uses scalable software containers. Suppose the number of messages to be processed increases suddenly during a high-impact event. In that case, the platform can use more containers to annotate them. SMDRM code is released as an open-source platform.[1] Its modules can be easily extended and adapted.

In the remainder of the paper, we describe the motivations for the platform development and the platform's architecture. We then describe the models that can be used within SMDRM, and one operational implementation.

2 Background

The usage of information from social media during emergencies has been one of the driving applications for research on the real-time processing of social media messages. Over a decade, research has sought to extract, categorize, and visualize relevant information for emergency management.

Floods have attracted significant attention for researchers, who have, for instance, attempted to determine flood extents using social media information, with some success. The uncertainty in the geolocation of messages has been reported as the main contributor to inaccuracies (Brouwer et al. 2017).

On average, the minimum time needed by emergency services such as the Copernicus Emergency Management Services (CEMS) Rapid Mapping (RM) service to provide crisis information after an activation request by an authorized user[2] is 24 hours (Wania et al. 2021). Furthermore, due to the technical issues with densely built-up areas, remote sensing analysis is of limited use in urban areas. These areas are commonly not analyzed and left out of the produced maps.

Previous research (Rufolo, Muraro, and Lorini 2021) shows that social media can provide a good overview of impacted infrastructures and provide situational awareness within a few hours. Social media postings immediately after the event have a higher probability of being relevant to

[1] https://github.com/ec-jrc/SMDRM

[2] EU Member States, EU Civil Protection Mechanism, the EC's Directorates-General and EU Agencies, the European External Action Service , as well as international Humanitarian Aid organizations

the event's detection and damage assessment process, and may contain less noise than later messages. This, according to practitioners, is helpful to crisis managers while waiting for EO products such as the ones by Copernicus Mapping[3].

3 Platform Description

3.1 Concepts

A **data point** is a dictionary, typically represented in JSON format, composed of a specific set of fields described in Table 1.

Field	Description
id	Unique identifier
created_at	The date and time at which the data point is created
text	The textual information to be annotated and/or geo located

Table 1: Mandatory fields for processing data points

To **annotate** is assigning a probability score to a data point's 'text' field. This is a float number between 0 and 1, representing the likelihood that the textual information in the 'text' field is of a specified category.

A Directed Acyclic Graph (DAG) represents a **workflow** of coded instructions, which we represent within the Airflow framework.[4] A DAG specifies the workflow as a set of repeatable coded rules, including dependencies between tasks, the order to execute them, and other instructions required to run a data pipeline.

A **task** is the smallest component of a pipeline. Each task must produce the same result every time it is executed on a defined dataset. It executes a specific logic, be it fetching data, running analysis, triggering other systems, or more.

SMDRM is a Python-based data pipeline application for processing social media data points. The goal of SMDRM is to provide an enriched version of the input data shown in Table 2 that can be further analyzed and visualized.

3.2 Scalability Requirements

SMDRM application is Docker Compose[5] based. A running Docker daemon and docker-compose software are required. Considering a minimal configuration intended to run on a single machine, the workstation minimum requirements are:

- 8 CPUs
- 12 GB free memory
- 10 GB free disk storage
- access to public docker registry

Suppose multiple servers are available, or SMDRM is deployed in a production environment. In that case, we recommend setting up an orchestrated solution that runs on several machines. In that case, Docker Swarm[6] may be the easiest way, as it is configurable via settings files.

[3] https://emergency.copernicus.eu/mapping/ems/cems-week-2021-conclusions-community-insights-and-service-evolutions

[4] https://airflow.apache.org/

[5] https://www.docker.com/

[6] https://docs.docker.com/engine/swarm/

3.3 Architecture

The main components of SMDRM are:

- Docker - ensures consistency, reproducibility, and portability across Operating Systems.
- Annotators - annotate disaster types and impacts, and writes information in datapoints.
- Geocoder - extract place names from text, looks for candidates in gazetteer and and writes information in datapoints.
- Apache Airflow - authors, schedules, and monitors workflows as Directed Acyclic Graphs (DAGs) of tasks in an automated, and distributed manner.

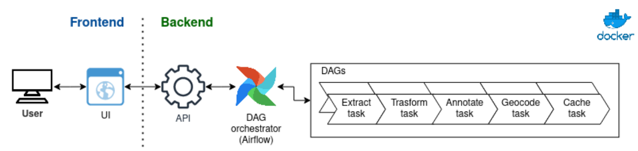

Figure 1: SMDRM architecture

The expected format of the input data is a zipfile archive. The zipfile should contain at least one Newline Delimited JSON (NDJSON) file. The NDJSON files must be located in the root of the zipfile archive. There must be a datapoint for each new line in the files compressed in the zipfile archive. The datapoint, including all required fields, can also be wrapped inside a ́tweet ́field. This is a template typically applied to keep the original record when data is transformed.

Field	Description
annotation	Annotation scores placeholder.
place	Geographic attribures placeholder.
place.candidates	place candidates returned by NER.
place.meta	Metadata of place candidates matched against gazetteer.
place.meta.city_name	Name of the city.
place.meta.country_name	Name of the Country.
place.meta.country_code	Alpha-3 code ISO for Country.
place.meta.latitude	Latitude of the place candidate matched against gazetteer.
place.meta.longitude	Longitude of the place candidate matched against gazetteer.
place.meta.region_id	The region identifier.
place.meta.region_name	The region name.
text_clean	Normalized textual information

Table 2: Fields added during data points processing

4 Filtering, classification, and geocoding

4.1 Filtering (E.g., Floods)

We use a supervised binary classification setup. The positive class comprised all messages indicating that a specific type of event (i.e. flood, wildfire, earthquake) is happening or is about to happen. Our work does not require semantic resources. We only leverage pre-trained encoders for multilingual modeling. They can be used to perform multilingual classification using embeddings that are aligned across languages(Conneau et al. 2017). We can use labeled data in a set of available languages to bootstrap a binary classifier for a new language for which no labels are available. In our case, we labeled data as relevant (label=1) or not(label=0) to floods or to water-related events (tropical cyclones). After several tests(Lorini et al. 2019), we decided to use LASER (Language-Agnostic Sentence Representations), released by Facebook[7], as a pre-trained language representation in multiple languages. LASER acts as the encoder in our model, as it provides the embeddings for the input sentences. So, we built a classifier network for our decoder in the model to classify the sentence as relevant or not to a specific event. Specifically, we trained a sequential model with two dense layers to minimize binary cross-entropy measure as loss function using the Adam algorithm as optimizer.

4.2 Categorization (E.g., Impact Assessment)

We aim to make the mining of social media messages useful for practical monitoring of urban events, building upon previous work in the JRC Unit around Social Media Flood Monitoring[8]. We noticed soon in our tests how the impacts generated from a water-related event are not different than any other disaster (i.e. injuries, evacuation, damages, services disruption) To obtain a high-quality training dataset, we performed a two-level annotation of impacts-related messages. The input to this annotation were tweets obtained during several flood or storm periods in the two cities for the pilot study. The annotators were the European VOST (Virtual Operations Support Team)[9] volunteers that process digital data for emergencies, usually composed of former or current members of various emergency response services.

Level 1: Impact / No impact The first level of annotation includes determining if a message describes an impact. In the instructions, we use the phrase 'negative impact' to avoid ambiguities in this regard and mention different types of impacts that can happen. However, We do not ask, for annotators to categorize messages based on those other types until the level 2 categorization is done.

Level 2: Type of impact The second level of annotation was focused on messages for which the level 1 annotation indicated they have an impact. According to our observations in the data, we considered various types of impacts that are common in urban events. First, we consider effects on specific individuals, such as people injured, missing, or displaced. Second, we split what is commonly referred to as the 'infrastructure and utilities' category into 'infrastructure damage' and 'service disruption'.

4.3 Geocoding

Depending on the aim of the application, the platform can sustain two levels of geocoding:

Regional level In our integration with EU-wide or World-wide monitoring systems, relevant messages are mapped to NUTS-2 areas (Nomenclature of Territorial Units for Statistics, Level 2). Geocoding deals with messages that do not include explicit geographical coordinates but mention a place name such as a landmark or city. SMDRM uses a Named Entity Recognition tagger to obtain possible locations in a text considering the syntax of the message. It then uses a gazetteer in an extensive database of place names with their corresponding geographical coordinates. Finally, it uses a series of heuristics to infer the correct country and correct gazetteer entry for those places. We use a library named DeepPavlov[10] for place names identification which extracts places candidates from a piece of text. Their coordinates and structured geographic information are then searched within a list of administrative areas and cities. Messages are aggregated at the level of an event but also at the level of each administrative area.

Urban level For the application of understanding urban flood impact, messages need to be geocoded at a level of granularity that is useful for emergency responders, which in this case needs to go into an intra-urban scale, i.e., they should refer to specific areas of a city which are affected by flooding events. Only a tiny fraction of the social media messages are geotagged precisely. For instance, only 1%-5% of the tweets are geotagged within urban areas(de Andrade et al. 2021). We used an innovative approach to achieve this goal. We focus on elements at risk of the infrastructure (such as a hospital, a factory, a school, or a stadium, among others), assuming that if a significant impact happens in such an infrastructure, at least some messages will mention the infrastructure by its name. However, this requires the creation of a customized gazetteer for infrastructure, which in turn requires an extensive database containing infrastructure elements' names. The database of known locations of infrastructure elements used in the project was built on data from OpenStreetMap (OSM)[11]. Infrastructure objects in OSM are identified through ' tags' that could relate to classes defined by the Sendai Framework indicators for Disaster Risk Reduction[12]

Figure 2 shows classified tweets aggregated by impact location and facilities. A manual analysis of the contents of the tweets has proved that many of the messages classified and geolocated seemed to indeed refer to impacts, some of them containing spatial references to entities which could be located close to the coordinates which were attributed to the tweets by the gazetteer.

5 SMFR, an Instance of SMDRM

Social Media Flood Risk (SMFR) is a platform to monitor specific flood events on social media (currently, only Twitter). The system is intended to work as a complementary monitoring service for existing early risk alert systems. The first release of this experimental project is tailored to work with EFAS[13], but in future releases, the 'topic' (floods, forest fires,etc) and the primary alert system will be configurable.

[7]shorturl.at/dwRXZ
[8]shorturl.at/bnNVX
[9]https://vosteurope.org/
[10]http://docs.deeppavlov.ai/en/master/features/models/ner.html
[11]https://openstreetmap.org
[12]https://bit.ly/3OFtnFt
[13]https://efas.eu

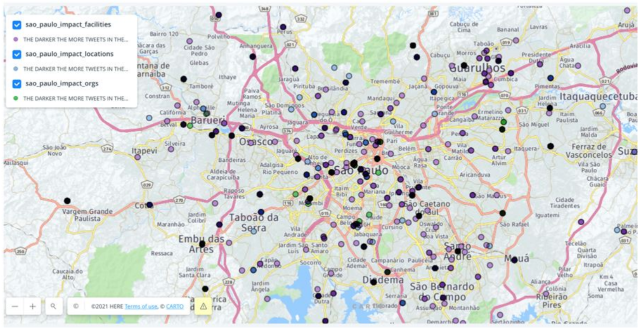

Figure 2: Relevant tweets aggregated by location/facilities.

For the deployment of SMFR we created a distinct DAG, and tasks tailored to the Twitter data structure. Each task iterates over batches of datapoints and applies a certain logic. The tasks are: **Extract:** enforces the SMDRM data structure onto each datapoint in a given dataset.

Transform: applies text normalization for the annotate task, and place candidate extraction via DeepPavlov NER model for the geocode task.

Annotate: annotate data through a multilingual model for the binary classification of tweets (flood relevance, float value ranging from 0 to 1).

Geocode: matches the place candidates extracted at transform task against the Global Places gazetteer.

Cache: saves processed datapoints from previous tasks into an Elastic Stack instance (Elasticsearch+Kibana) to enable data exploration/visualization in dashboard style.

Finally, we created an additional DAG, and tasks specific to the creation of products sourcing EFAS interfaces. The DAG produces GeoJSON products representing areas and risk grade, most relevant tweets per reported area, and trends per day. These files are disseminated to a list of map servers. If an area presents less than ten highly-relevant tweets (relevance > 0.8), the associated region is Gray. A region is Orange if the ratio between medium-relevant tweets ($0.2 <$ relevance < 0.8) and highly-relevant tweets is between 5 to 1 and 9 to 1. Region is Red if the ratio exceeds 1 to 9. After deduplication, the 5 most relevant tweets are selected and presented for each area. A product is visible in Figure 3

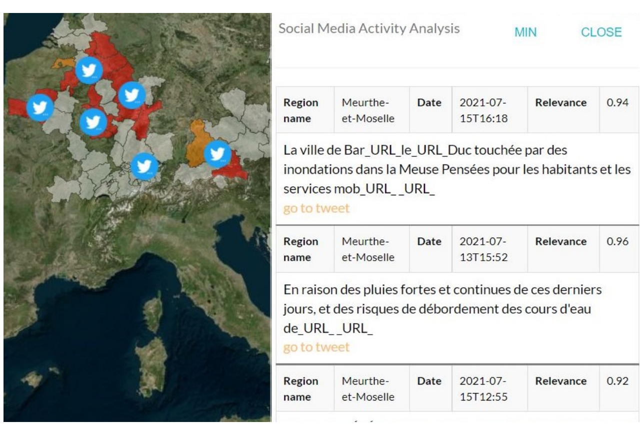

Figure 3: Screenshot from SMFR.

6 Future Work

One ongoing development is the implementation of a Representational State Transfer (REST) Application Programming Interface (API). This represents the entry point where the end-user can leverage the functionalities of the SMDRM platform via HTTP requests. Components such as classification models for annotation or DAG runs can be executed through dedicated API resources.The platform allows the analysis of texts and images and we kept it independent from Twitter-specific fields. Therefore a future development could be to extend the source of data to other social networks or crowdsourcing platforms. Another challenge that we are tackling is improving geocoding of messages collected during urban events. We are trying to detect partial matches between facilities and mentions in the text. As mentioned in Section 1 we trained a neural network for images classification for 5 types of disaster using the EfficientNet model structure. Image classification is currently performed by interacting with the platform asynchronously, not supervised by the Airflow component but by running offline scripts. One work in progress is to develop a module for handling media processing. We should also define a quantitative analysis of actual vs. predicted values extending the analysis to all tweets available for some test cases and identifying missing information. In this way, we could also compute values for recall analysis.

References

Brouwer, T.; Eilander, D.; van Loenen, A.; Booij, M. J.; Wijnberg, K. M.; Verkade, J. S.; and Wagemaker, J. 2017. Probabilistic flood extent estimates from social media flood observations. *Nat. Hazards and Earth Sys. Sci.* 17(5): 735–747. doi:10.5194/nhess-17-735-2017. URL https://nhess.copernicus.org/articles/17/735/2017/.

Conneau, A.; Lample, G.; Ranzato, M.; Denoyer, L.; and Jégou, H. 2017. Word Translation Without Parallel Data. *arXiv preprint arXiv:1710.04087* .

de Andrade, S.; De Albuquerque, J.; Westerholt, R.; Restrepo-Estrada, C.; Morales, C.; Mendiondo, E.; and Delbem, A. 2021. The effect of intra-urban mobility flows on the spatial heterogeneity of social media activity: investigating the response to rainfall events. *Int. J. of Geo. Inf. Sci.* doi:10.1080/13658816.2021.1957898.

Hutchins, E. 1995. *Cognition in the Wild*. MIT press.

Lorini, V.; Castillo, C.; Dottori, F.; Kalas, M.; Nappo, D.; and Salamon, P. 2019. Integrating Social Media into a Pan-European Flood Awareness System: A Multilingual Approach. *Proc. of the 16th Int. Conf. ISCRAM* URL http://idl.iscram.org/files/valeriolorini/2019/1854_ValerioLorini_etal2019.pdf.

Rufolo, P.; Muraro, D.; and Lorini, V. 2021 (KJ-NA-30632-EN-N (online)). ISSN 1831-9424 (online).

Wania, A.; Joubert-Boitat, I.; Dottori, F.; Kalas, M.; and Salamon, P. 2021. Increasing Timeliness of Satellite-Based Flood Mapping Using Early Warning Systems in the Copernicus Emergency Management Service. *Remote Sensing* 13(11). ISSN 2072-4292. doi:10.3390/rs13112114.