Filtering Informative Tweets during Emergencies: A Machine Learning Approach

Flavia Sofia Acerbo
Politecnico di Torino
Torino, Italy
flaviasofia.acerbo@gmail.com

Claudio Rossi
Istituto Superiore Mario Boella
Torino, Italy
rossi@ismb.it

ABSTRACT

Thanks to their worldwide extension and speed, online social networks have become a common and effective way of communication throughout emergencies. The messages posted during a disaster may be either crisis-relevant (alerts, help requests, damage descriptions, etc.) or not (feelings, opinions, etc.) In this paper, we propose a machine learning approach for creating a classifier able to distinguish between informative and not informative messages, and to understand common patterns inside these two classes. We also investigate similarities and differences in the words that mostly occur across three different natural disasters: fire, earthquake and floods. The results, obtained with real data extracted from Twitter during past emergency events, demonstrate the viability of our approach in providing a filtering service able to deliver only informative contents to crisis managers in a view of improving the operational picture during emergency situations.

CCS CONCEPTS

• **Computing methodologies → Information extraction**; **Machine learning**; • **Applied computing**;

KEYWORDS

Machine Learning, Social Media, Emergency Management

1 INTRODUCTION

Due to climate change, the number of natural disasters has largely increased in the last few years. At the same time, during such events, it has become very common to share information on social networks. This information may be useful for emergency managers who must take focused actions in the shortest time possible. However, social media streams are full of useless information, and this limits their use during crises due to the impracticability of manually checking each content in search of actionable information. Hence, an automatic tool aimed at discarding useless (not informative) content could greatly reduce the amount of content to be analyzed, making social media monitoring during crises an easier task.

In this paper, we evaluate the viability of using a machine learning approach to classify social media posts according to their informativeness.

We test our approach on real data collected from Twitter, a very popular microblogging platform, during several natural disasters. We exploit both Twitter-specific features together with text analysis to define the classifier feature set. We define a novel text metric and we test it using different approaches, including recent word-embedding techniques.

The classifier performance demonstrate that our approach is viable for being operationalized into a service able to extract informative contents from Twitter streams related to the considered natural disasters (fire, earthquakes and floods).

2 TWITTER

Twitter is a social network platform that has largely grown during the last few years and that is now counting around than 330 millions active users per month. Twitter is a micro-blogging service, which means that users can write messages with length up to 140 characters: these messages are called *tweets*. A user has *followers*, i.e., users who follow him/her, and *followee*, i.e. users that he/she follows. Relevant users, such as Governments or public personalities can be verified by Twitter. Words followed by # are called *hashtags*, and they are used as tags in the text. Other users can be referenced in a *tweet* using the character @ followed by their usernames. Thank to *retweets*, users can share messages that were already written and/or shared by other users, allowing for message propagation in Twitter.

3 RELATED WORKS

Despite being a relatively recent field of study, interesting works have been done around the use of social media networks within emergencies. Olteanu et al. [?] present the result of a manual labelling campaign to describe what to expect from social media data across a variety of emergencies (natural disasters, terrorist attacks, explosions, etc.) in terms of volume, informative level, type and source. Castillo [?] reviews how to use Big Data generated by social media platforms during emergencies, underlining the major challenges across several dimensions, including volume, variety, velocity, validity, visualization, and value. Other studies [?] analyze the reliability of viral *tweets* and the retweet activity, showing how the propagation of rumors on Twitter differs from the one of real news. Jackoway et al. [?] extract event-related *tweets* using information coming from news articles, but do not look at the informativeness of a *tweet* and do not consider unexpected events such as earthquakes. Klein et al. [?] propose a Natural Language Processing approach coupled with a clustering algorithm to tag *tweets* as related to an emergency event or not, while Caragea et al. [?] compare several approaches in comparison to Bag of Words to classify text messages written during the Haiti earthquake and gathered by the Ushahidi platform [1] into different information classes. The closest work to ours, is the one presented by Longhini in his master thesis [?], who proposed a classifier for informativeness using text-agnostic approach that used Twitter specific features without considering the text of *tweets*. Our work aims to realize

[1]https://www.ushahidi.com/

a further step: define text-based features to improve the classifier performance while minimizing language-dependent components.

4 METHODOLOGY

We create a Machine Learning model using the data chain illustrated in Figure 1. We assume to have a dataset in which each *tweet* has been manually labeled into two classes (informative and not informative). This is a common assumption in all supervised machine learning approaches. Similarly to [?], we define the concept of informativeness as everything that can be useful to improve the situational awareness for both citizens and authorities about an emergency event. Given such specific definition of the informativeness concept, an ad-hoc model is required to perform the classification task.

To avoid costly and computationally heavy grammatical analysis tools, and to minimize the language dependent components of our solution, we chose a word segmentation approach to analyze the text of *tweets*. First, we perform common pre-processing steps: remove punctuation (which includes also the character #), stopwords, words shorter than 3 characters, numbers, mentions to other users (i.e. @username), URLs. We transform everything to lowercase and we remove all words used to query Twitter because they appear in almost all *tweets* of the dataset, hence they are not correlated with the informativeness. Instead of training different models for each language, we translate each word (word-by-word translation) into English before to proceed with the following steps. This is both to extend the training data, and also to makes the methodology applicable worldwide, provided that a translator is available. [2] We aggregate words having the same meaning using two different methods: stemming and lemmatization. For the latter, we remove all words whose lemmas are unknown and the ones recognized as proper nouns. We observe that words with unknown lemma are often typos, location names, or hashtags made by event-specific concatenated words, while proper nouns generally refer to places where the event happened. We remove those words because we aim at implementing a location independent solution that could be applied to any event of the same type. Note that the identification of the *tweet* language is not required for implementing an operational solution, because the *tweets* should be retrieved for a given language-topic pair.

We perform a Document Term Matrix on the initial dataset, taking each *tweet* as a document, and for each event type under study (e.g. floods, fire etc.) we create two clusters of words, where each cluster contains only words belonging to *tweets* homogeneously classified (informative, not informative) and their term frequency. We create a couple of cluster for each event type because the lexicon is specific to the event. As detailed in Section 4, we use the clusters to compute our text metrics, while we use the Twitter APIs[3] to extract social media specific features.

As in previous works on automatic classification of *tweets* aimed at detecting spam users [?], predicting viral *tweets* [?], and classifying the credibility of a *tweet* [?], we propose to use of text-agnostic features . As shown in Table 1, we select features referring both to the *tweet* and to the user who generated it. We also consider the features *Source* and *Retweet* designed in [?], because of their effectiveness. The Source feature is a discard rate calculated on how many not informative *tweets* were created from each source, where the source is the application used to generate the content (e.g. Twitter Web Client, Twitter for iPhone, etc..). Instead, the *Retweet* feature is a *retweets* count over a fixed temporal window.

We define our text features using two different approaches:

- **Linear Similarity**: Let N(t) be the number of words in the *tweet* (t), f_I(p) and f_{NI}(p) the frequency of the word p in the Informative and Not Informative clusters, respectively. The informative and not informative Text Metric (TM) of the *tweet* (t) is given by:

$$TM(t) = \sum_{i=1}^{N(t)} f_I(p_i(t))$$

$$\overline{TM}(t) = \sum_{i=1}^{N(t)} f_{NI}(p_i(t))$$

- **Word2Vec**: Let $W(p) = w \in w2v_n(p)$ be the set of the top n most similar words to the word p resulting the Word2Vec ($w2v$) embedding, and $cos(p1, p2)$ the cosine similarity between the words $p1$ and $p2$, then the informative and not informative Text Metric of the *tweet* (t) is given by:

$$TM_{w2v}(t) = \sum_{i=1}^{N(t)} \sum_{j=1}^{n} f_I(p_j(t)) \cdot cos(p_i(t), p_j(t))$$

$$\overline{TM}_{w2v}(t) = \sum_{i=1}^{N(t)} \sum_{j=1}^{n} f_{NI}(p_j(t)) \cdot cos(p_i(t), p_j(t))$$

We train machine learning classifier using all obtained features to generate our model.

The trained classifier can be used to filter *tweets* to retain only informative ones once an event is detected. Note that (i) the methodology applies also to a monitoring scenario, in which decision makers wants to filter informative content on a given topic; (ii) the search query to extract test data should contain at least the same words used for extracting the training dataset.

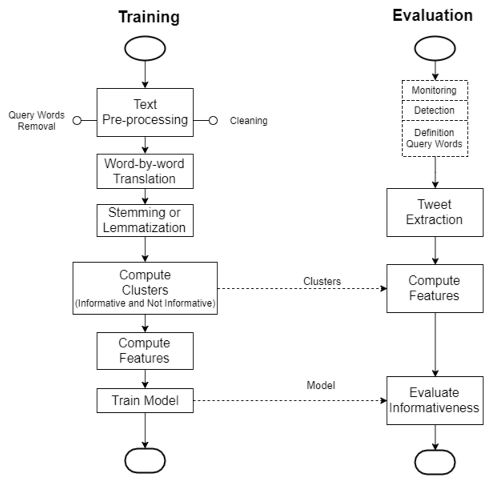


Figure 1: Overall diagram of the proposed solution.

[2]Having an offline translator would avoid additional operational costs.
[3]https://dev.twitter.com/rest/public

Table 1: Base (text-agnostic) feature set

Feature	*Description*
Delta Time	Time interval from the beginning of the crisis and the time the *tweet* was posted
Followers	Number of user's Followers
Followee	Number of user's Followee
Registration Date	Time interval from the registration date of the user
Total Tweets	Total number of *tweets* posted by the user
Verified User	Boolean value: is the user verified?
GeoTag	Boolean value: is the *tweet* geotagged?
Hashtags	Number of Hashtags in the *tweet*
Links	Number of URLs in the *tweet*
Mentions	Number of @ in the *tweet*
Source	Discard Rate of the source of the *tweet*
Retweet	Number of *Retweets* reached in the first 10 minutes after the posting of the *tweet.*

5 IMPLEMENTATION

To train and test our model we use a large dataset of *tweets* collected by Olteanu et al. [?] during 26 emergencies occurred in years 2012-2013, from which we kept only the ones related to natural hazards having more than one event in the dataset. The considered event list together with the number of available *tweets* is shown in Table 2. Note that this dataset was labeled using the same informativeness definition as introduced in Section 1. In order to properly apply our methodology, we determine the language of each *tweet* in the dataset [4], because the *tweets* were collected without any language filter. In the considered dataset, each *tweet* there is an informative label, which can assume 4 different values:

- *Related and informative*: the message is about the event and it contains some useful information.
- *Related - but not informative*: the message is about the event, but it does not contain any useful information.
- *Not Related*: the message does not talk about the event.
- *Not Applicable*: the message is not comprehensible.

Because our focus is the evaluation of the *tweet* informativeness, and not its relation with the event, we exclude all *tweets* labeled as *Not Related* or *Not Applicable.*

Similarly to [?], we select a temporal windows of 10 minutes, and fit a log-normal distribution to obtain the timestamp of the *retweets* that are not returned by the Twitter's APIs, which is limited to the most recent 100 *retweets.* Note that the log-normal distribution has been proven to be a reasonable fit for the *retweet* activity [?].

Considering the *tweet* ID and using the Twitter REST APIs[1] we retrieve all information needed to create the Baseline Features.

We write all code with the *R* programming language and use the following libraries:

[4]we use microsoft Azure cognitive services to do so

Table 2: Events considered from the CrisisLexT26 Dataset.

Event	*Year*	*Number of Tweets*
Alberta Floods	2013	5887
Philippines Floods	2012	2950
Manila Floods	2013	2032
Colorado Floods	2013	1778
Queensland Floods	2013	1223
Sardinia Floods	2013	1143
Italy Earthquake	2012	7351
Guatemala Earthquake	2012	3261
Bohol Earthquake	2013	2214
Costa Rica Earthquake	2012	2193
Colorado Wildfire	2012	4172
Australia Bushfire	2013	1982

- **tm**: package used for text mining, such as corpus handling, pre-processing, stemming and generation of Document Term Matrix. [?]
- **caret** (**c**lassification **a**nd **re**gression **t**raining): package used for Machine Learning that contains methods to train and test the most common algorithms such as Random Forest (the one we have chosen), Neural Networks, Support Vector Machine (SVM) and more. [?]
- **koRpus**: package used to perform lemmatization, and more generally text analysis [?]. It relies on a third party software called *TreeTagger* which annotates text with part-of-speech and lemma information.[2]

[1]https://dev.twitter.com/rest/public
[2]http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

6 CLUSTERS CHARACTERIZATION

In this section we comment the results obtained, both in terms of clusters and performances of the trained classifier. We show in Figures 2 and 3 the wordclouds plotted according to words relative frequencies, containing the 50 most frequent words of the informative and not informative clusters, respectively. By means of a qualitative analysis, we can make some considerations about the different nature of the hazards, people needs, and the limitations of our approach.

What immediately stands out is the similarity between all the not informative clusters: it is clear that in all types of events most not informative *tweets* contain emotional aspects, feelings and religious content such as prayers, hopes, thoughts, gratitude and support. Of course, any of this personal and psychological information, which is very common on social network during a disaster, is not useful for decision makers. In addition, we find frequent common adjectives in these clusters (e.g. good, bad, great etc.).

Looking deeper at the informative clusters, we can see a very close similarity between the fire and the flood clusters, and a clear difference with respect to the earthquake one. This is probably due to the different phenomenology of such events. While fires and floods may be preceded by weather related alerts, earthquakes are not predictable and strike fast. Floods and fires have a longer response phase (event ongoing), which gives time for messages of help, requests of volunteering or descriptions of damages. Conversely, an earthquake usually goes off very rapidly, so messages often refer only to consequences (which could be an incoming tsunami), death tolls or damages.

It is interesting that fire messages contain words related to the amplification forces (wind, air), while during a flood the most frequent content is about the causes that generated the emergency (water, storm, rain).

We note that many words are in common between the informative cluster and the not informative one, even if with different frequencies. This may generate a "noise" on our model when trying to give two different scores of informativeness and non-informativeness. To mitigate this problem, we delete all common words from the cluster having the lowest frequency and to subtract it from the highest frequency. The word is deleted from both clusters in case of ties.

Also, location names having other meanings are still found in the clusters. For example, the word *wale* is present in the informative Fire Cluster. It comes from the *tweets* of the Australia Bushfire, which happened in New South Wales. Since the lemmatizer we used is not able to recognize geonames made by more than one word, *wales* is kept as the plural of wale. Always in the Informative Fire Cluster, we find *blue* that comes from the Blue Mountains (Australia), where one crisis happened. In the Informative Earthquake Cluster, instead, we find *chile*, which in the original text referred to the Southern American state, but that was interpreted by the lemmatizer as the American spelling of the word *chili*. Such words are not linked with informativeness, and should be removed from the clusters. The automatic disambiguation of location names is beyond the scope of this paper and it is left as future work.

7 RESULTS

In order to test our classifier in a realistic and conservative setting, we evaluate its performances using the *Leave One Out* approach, i.e., we perform one iteration for each event, excluding it from the training and the word cluster computation, while using it for testing. Although we tried several algorithms (e.g., bagging, boosting, neural networks, SVMs), we use the Random Forest classifier with default parameter, which is the one that gave us the best performances. First, in Figure 4, we compare the classification accuracy obtained with stemming and lemmatization, considering all features described in Section 1 and for each event type. We select lemmatization over stemming due to its higher performance, i.e., +3.8% accuracy, on average.

We assess the impact of translating all words into English before clustering in Figure 5. Unexpectedly, the translation improved the accuracy by 7%, on average. This is due to both the aggregation of synonyms, which works as an aggregation function, and to the higher representativeness of clusters, which are computed with a higher number of elements per cluster.

Table 3: Most similar words to "*floods*"

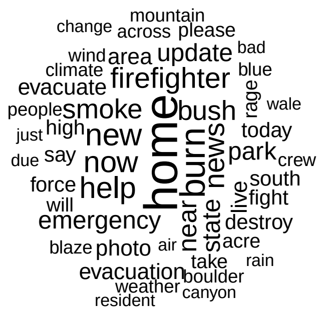
Words	*Cosine Similarity*
floods	1
flooding	0.8720026
mudslides	0.8089744
landslides	0.8013214
storms	0.8012049
rains	0.7763067
torrential	0.7651205
droughts	0.7541500
downpours	0.7218243
heatwaves	0.7046358

Using Recursive Feature Selection we select the best combination of features to be used, which are the first 7 of the ranking shown in Figure 6. Note that the text metrics have the first two places in the ranking, with a big difference in importance (computed by the Random Forest Model) with respect to the third one, which is the Source discard rate. This proves that there is an high correlation between the lexicon used within the same type of emergency. As shown in Figure 7, adding the Informative Text Metric to the Base feature set improves the accuracy by 4.7%, on average, while adding the Not Informative Text Metric further enhances it by 3.5%, on average. The RFE selection slightly improves with respect to the full feature set.

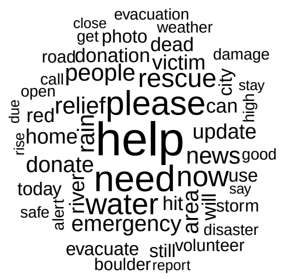
In Figure 8 we show the confusion matrix for the RFE features set taking all tests and averaging the results. Considering the Informative Class as the Positive Class, we can see that the *True Positive* rate is very high, while we have poorer results in the *False Positive* rate. The resulting F-Measure is 0.77. Even if the Delta Time feature can be used only after the occurrence of an event, we checked that its substitution with the Followees feature carries negligible changes in the performance. Finally, we evaluate also the Word2Vec approach, computing the two $\mathrm{TM}_{\mathrm{W2V}}$ taking the 10 most similar word. Word2Vec is a word embedding model created by Mikolov et al. [?] that is used to learn a vector representation of words. It is a neural network that takes a large corpus as input and produces as output a vector space commonly with 100-300 dimensions, where each vector is a different word. The more two vectors are close to each other, the more the two words that they represent will be similar. Using *cosine similarity*, no similarity is seen as 0, total similarity as 1, while other cases will have a coefficient between 0 and 1. For example, the most similar words to the term *floods* are shown in Table 3 together with their cosine similarities. This is obtained by training the W2V model with the English corpus *1 Billion Word Language Model Benchmark*[5], produced from a News crawl data.

As shown in Figure 9 the accuracy with W2V is lower by 4.2%, on average. The performances could be improved with enriched clusters, i.e., considering more events to compute them, and by training W2V with a Corpus more specific on natural hazards.

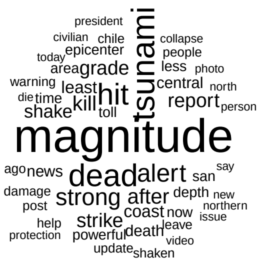
[5]http://www.statmt.org/lm-benchmark/



(a) Fire

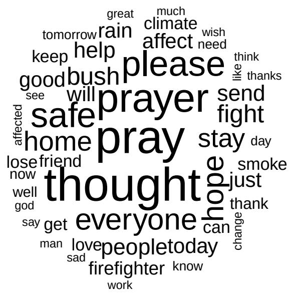


(b) Floods

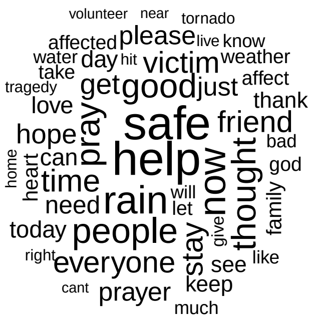


(c) Earthquake

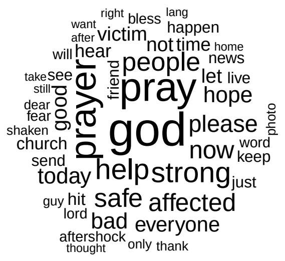
Figure 2: Top 50 words of Informative Clusters.



(a) Fire



(b) Floods



(c) Earthquake

Figure 3: Top 50 words of Not Informative Clusters.

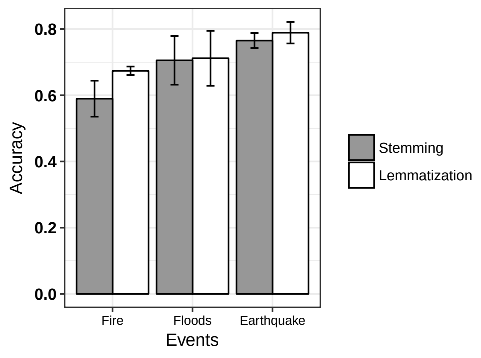


Figure 4: Accuracies and standard deviations for stemming (0.69 accuracy) and lemmatization(0.73 accuracy).

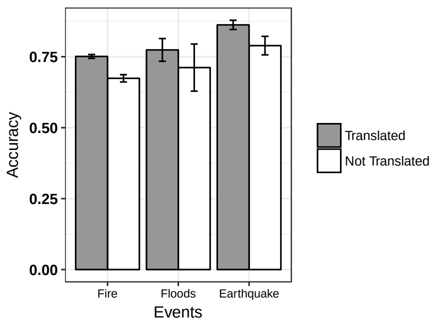


Figure 5: Accuracies and standard deviations using the lemmatized text in its original language (single cluster for each combination of [language, event]) or translated having a single English cluster for each event). Mean accuracy of 0.73 and 0.8 for original and translated, respectivelty.

8 CONCLUSION AND FUTURE WORKS

The classifier we implemented is capable of filtering *tweets* related to floods, fires, and earthquakes according to their informativeness with an overall accuracy of 76%. The best performances are achieved coupling text-agnostic features taken from literature with our Text Metric. Our approach can be applied for any event among the evaluated ones, in any location and language, provided that a lemmatizer and a translator are available for all required languages.

Future works will include the disambiguation of location names to improve the quality of the clusters, the extension of the training data to a wider dataset, the test of other word embedding techniques and Corpora, and finally the creation of a new classifier able to discriminate among several information classes.

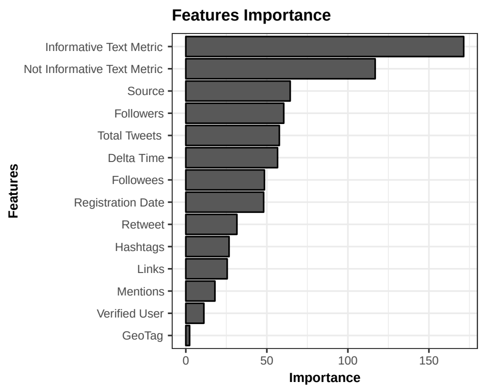


Figure 6: Features ranked by importance.

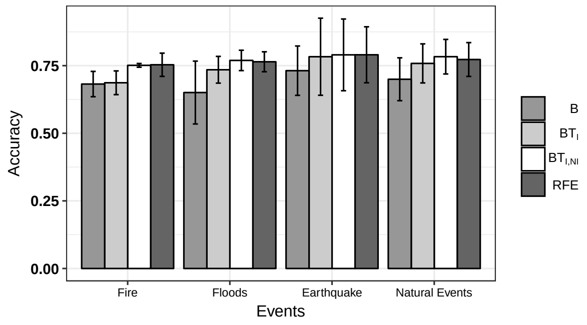


Figure 7: Accuracies and standard deviations for different combinations of features: B (Base features), $\mathbf{BT_I}$ (base + $TM(t)$), $\mathbf{BT_{I,NI}}$ (base + $TM(t)$ and $\overline{TM}(t)$) and RFE.

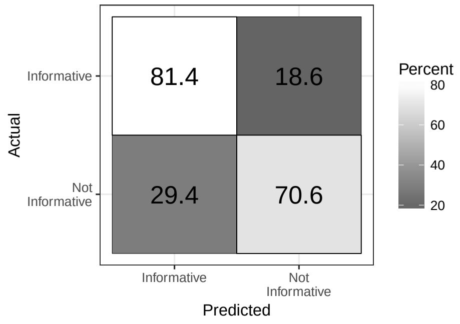


Figure 8: Confusion matrix obtained using RFE and by averaging the performances of all tests. F-Score: 0.77

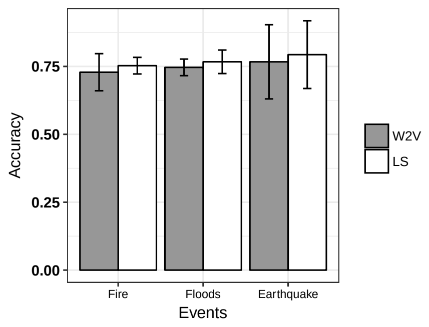


Figure 9: Accuracies and standard deviations using the Linear Similarity and the Word2Vec Text Metrics.

ACKNOWLEDGMENT

This work was partially supported by the European Commission through the H2020 I-REACT project, grant no. 700256.