Classification of Crisis Tweets

Report by Abbas Keshavarzi, Vyom Shrivastava, Dong-Yu Yu, Manoj Raghorte

Introduction

Since the advent of social media, it is widely used in everyday life but it's importance is imperative in events of crisis. Researchers have been using social media content to analyze the impact and severity of emergency situations but the most useful case is using social media to identify people who are experiencing real threats of crisis in current time and managing resources to rescue them or atleast curtail their damages. These use cases are not only applicable in mass catastrophic events but also emergency situations which impacts small group of individuals.

While analyzing social media especially twitter information in cases of emergency, there are key challenges to deal with.First, it is hard to find the actual information which can depict the real picture of crisis. In case of twitter, many users just broadcast information without enough knowledge of emergency or information about their location or sometimes just retweets.As a result, from all the pool of user's, it becomes hard to identify users who actually need help and so necessary resources can be allocated to them. After extracting enough geo information, it equips us with handful of information which needs further purification for actual operational use.

We propose a classification system of crisis tweets which identifies if the provided tweets belongs to emergency case or non emergency case. This classification of tweets can be viewed as first layer of purification which differentiates actual necessary tweets from the pool of mass information which can be used by researchers or emergency help teams for operations of actual help.

Original contributions:

In this work, we implement a web application system to classify crisis tweets that can be accessed by users on internet to differentiate crisis related tweets from others. In this work, we also provide a batch system option to users who might want to use our work to apply for huge set of tweets which they might have in hand. Unfortunately crisis happens around the world each day and this type of work would be good assist to identify peoples who are in emergency and dire need of help. Although we test our system on Tweets in English, it can be easily be modified to be applicable to other languages with minimum effort. We can name another characteristic as our unique contribution. We include different tweets from different crisis into our database as it makes our dataset unique in a sense that no other work in this area has such broad variety of tweets in this domain. And it helped us to have more powerful classification models as we will discuss in further sections.

Broader impact

In today's world, there is no end to emergency situations happening almost everyday. Though there might not be mass emergency events happening everyday but some part of the world definitely suffers some kind of crisis at any given time. Acknowledging the popularity and use of twitter worldwide, our application can be used not just by researchers and scientific community but also by ordinary people, government agencies as well as relief organisations. We have extended this application which allows all kinds of emergency tweets and further advancement can be done for automated use of this application as well as real time utilization also.

System architecture:

The system we proposed contains three separate modules. The first module is the extracting module, which is responsible to gathering related tweets to crisis. This part can be done online or offline. By offline we meant that we are able to retrieve tweets which published in past. As we or other users may have access to some dataset with tweet ids and not the whole tweets and its information, this module is able to process such dataset and retrieve all of them. On the other hand, by online means, as we did in previous part of this project, we can extract original tweets in the time of crises which we call it online extracting process. The second part is the model generator module. The dataset provided by the extractor module is the input to model generator and after pre-processing phase, we can build different models based on different algorithms and provide them for the next phase. We will take about this phase in the next section as more detail to our approach is required to fully understanding this module, but it is worth to mention that one advantage for our system is that we do not stick to one algorithm or one approach, and this flexibility can be easily provided to end user, which we skip it and leave for future work as the time was limited for this project. The last part of our system is the web application. To have more user-friendly platform to all and for testing purposes for this project, we build a simple web application using PHP and JavaScript that provide a service for users to check whether a tweet is crisis related or not. For this web application it is enough to get a tweet or any string and by run it through our models, we will notify our prediction to the user. In figure 1. We tried to simplified our structure to main components describe above to get reader a high level idea of our architecture.

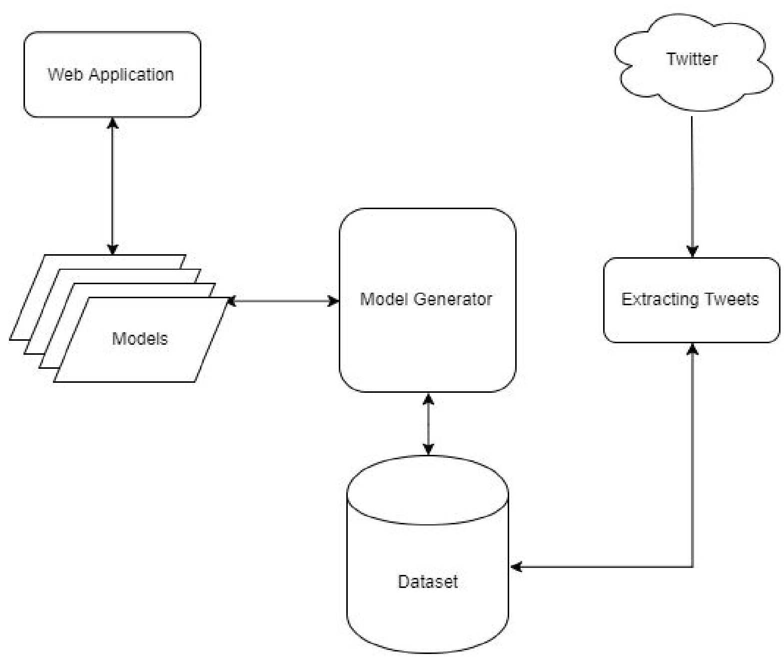

Figure 1. System Architecture

Technical details:

Our project details can be summarized in the following order:

Data-Collection:

We used the dataset from Crisis NLP website, we used the dataset from **[1]**, the dataset consisted of twitter IDs of tweets from various different crisis. We extracted the tweets using the twitter ID and created a single dataset.

Data Annotation:

The Crisis NLP dataset was already annotated but in various different categories, we changed the annotations to only 2 categories i.e. emergency and non-emergency

Pre-Processing:

1. We first cleaned our data by removing special characters ($ # / ” ’ . &). We then also got rid of the special ASCII symbols in data i.e. emojis.
2. Then we removed the stop words like I, the, can, for, is, an etc.
3. Next we did stemming of the words, i.e. each word was changed back to its root word.
4. As the machine learning algorithms do not understand words but only numbers, we changed the tweets in vectors using Count Vectorization and Tf-idf.

Multi-nomial Naive Bayes:

Multi-nomial Naive Bayes is one of the most successful algorithms in the field of text classification. It is based on the Bayes theorem which is mentioned below:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Here, B is data and A is the class. The algorithm tries to predict the probability of any feature B to be in the class A using previously known information.

Support Vector Machine:

SVM is another popular tool used in text classification, which tries to find the hyper-plane to distinguish the linear separable datasets, which makes it very suitable for the text classification tasks. Also, in our work, we try to classify the tweeters into two subsets, “emergency” and “nonemergency”, which in nature is more efficient to deal with SVM. We used SGDClassifier and set the loss parameter equal to hinge, which gave us an efficient SVM classifier to use.

Experiments:

We combined the above two algorithm with three pre-processing tools, Count Vectorization, Tf-idf, and pipeline of CV and Tf-idf, to form our models. We randomly picked 80% in our dataset for training, and the remaining for testing. The results are shown in Fig. 2. Since the features in both dataset are similar, the error rates between training and testing are quite close. Among those models, SVM combing both pre-processing tools achieved highest accuracy rate. Regarding to the ability to capture the true positive cases, Multi-nomial Naive Bayes achieved highest with 285 cases. Because in real-time detection, we want to increase the tendency to identify as many emergency events as possible considering the high cost of missing one, we chose the latter for usage in our user interface.

	Naïve Bayes CV	Naïve Bayes tf-idf	Naïve Bayes both	SVM cv	SVM tf-idf	SVM both
Training acc.	84.00%	81.65%	80.99%	82.50%	82.03%	81.75
Testing acc.	83.60%	80.97%	80.60%	83.83%	83.38%	84.06%
TP	285	164	148	269	235	242
FN	96	217	233	112	146	139
TN	827	913	924	846	874	876
FP	122	36	25	103	75	73

Figure 2. Comparison of different models

Team Members & Responsibilities:

1. **Abbas Keshavarzi:** Created front end Web Application, Handled Testing of models.
2. **Manoj Raghorte:** Data extraction, data cleaning and manual annotation.
3. **Vyom Shrivastava:** Data Preprocessing, Created and trained Multinomial Naive Bayes models.
4. **Yu Dong-Yu:** Created and trained SVM models, Created pipeline for combined models.

Related work

There has been many applications which classifies tweets which would be doomed as useful or not. Many applications are used only in context of research and not extended for use of ordinary people. Basically, all those applications are machine learning models and learning algorithms used are mostly Naive Bayes and Support vector machine and classified tweets are evaluated across another dimension of metrics such as accuracy and precision measure of tweets. We extend the previous work by creating a web application which can identify tweets across all emergency situations such as hurricanes,earthquakes, wildfires etc and not just a single domain of crisis. Also we used six individual classifier models with different preprocessing to each model which evaluates tweets on different parameters each.

Conclusion:

In this work, we examined popular text-mining models based on Multi-nominal Naive Bayes and Support Vector Machine. We imported pre-processing techniques such as Count Vectorization and Tf-idf to increase the accuracy of our models.The best one, SVM with both pre-processing tools, in our models can identify the unseen dataset with 84% accuracy, while Naive Bayes with both pre-processing tools can capture more True Positive twitter. We contribute on the valuable domain of crisis identification, which helps authority with the capability to quickly classify the importance on the Twitter and focus and allocate resource on the most urgent tasks. In the future, we want to think about the possibility to combine all the models we have to best satisfy the nature of this domain.

References:

[1] Muhammad Imran , Prasenjit Mitra , Carlos Castillo; Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages; In Proceedings of the 10th Language Resources and Evaluation Conference (LREC), pp. 1638-1643. May 2016, Portorož, Slovenia.

[2] Kibriya A.M., Frank E., Pfahringer B., Holmes G. (2004) Multinomial Naive Bayes for Text Categorization Revisited. In: Webb G.I., Yu X. (eds) AI 2004: Advances in Artificial Intelligence. AI 2004. Lecture Notes in Computer Science, vol 3339. Springer, Berlin, Heidelberg

[3] Joachims T. (1998) Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398. Springer, Berlin, Heidelberg

[4] Crisis NLP Dataset. http://crisisnlp.qcri.org/lrec2016/lrec2016.html