Real Time Twitter based Disaster Response System for Indian Scenarios

KRISHNA KANTH A, ABIRAMI S, CHITRA P, GAYATHRI SOWMYA G
Department of Computer Science and Engineering,
Thiagarajar college of Engineering, Madurai.
krishna@student.tce.edu, abirami.2t@gmail.com, pccse@tce.edu

Abstract— *Twitter is a popular social media platform with more than 1 million daily active users. Mostly, all breaking news is posted earlier in twitter than any mainstream media. Hence, this micro-blogging social network experiences a deluge of information flow during natural disasters. Situation based mining of information from the twitter data, can play a significant role in disaster response and recovery. The large volume and velocity of data flow on twitter during disaster makes it tedious for the disaster rescue volunteers to manually analyze and retrieve information from them. An automated system that could retrieve relevant information from this enormous twitter data during a disaster, could be useful for the disaster relief volunteers to accomplish their duty efficiently amidst the chaos. During disasters, the volunteer's team may service more efficiently, if they had a classification based on the victims who request for donation and those who request rescue. In this paper, we propose an artificial intelligence based real time disaster response system-Disastro, which assists the volunteers by identifying the relevant tweets from the real time twitter data and classifying them under the domains "rescue" and "donation". Disastro is empirically validated across various machine learning algorithms for classification using the tweets posted during Chennai rains and Kerala floods. The versatility of Disastro across different disasters and its improved classification accuracy makes it flexible and robust to handle any location-based emergencies.*

Keywords: Twitter data, disaster response system, artificial intelligence, machine learning, and classification.

I. INTRODUCTION

Natural disasters are unexpected events that cause extreme loss of lives and properties. In order to lessen the severity of the event and promote resilience, it is mandatory to provide accurate information to the rescue volunteers about the needs of the victims. This can be achieved through a disaster response system that could retrieve and render accurate information to the volunteers Twitter is a free micro-blogging service that allows users to create and share short messages. This also enables them to send a brief and timely description of the happenings precisely. Partial availability of Twitter data provides officials and academics with the opportunities to gather public tweets and retrieve information from them in response to user-defined triggers and specific queries [4]. Among the available social media platforms, Twitter has immense potential to serve as an additional information layer to the current emergency response systems and there is a rich and growing body of research to support this [1,2,3].

Some of the related works include AIDR [5] where the author incorporates artificial intelligence for developing a disaster response system. Here, the tweets are labelled manually using the labels informative, true damage, casualties, and donation. A machine learning classifier was trained on the annotated dataset and deployed in the response system. Through this system, the authors concluded that automatic classification using pre-existing training data would not be a satisfactory common solution across all types of crises as their domain adaptation was difficult. Tweedr [6] is another disaster response system that utilizes data mining tools and machine learning based classification model for mining twitter data and identifying tweets reporting casualties and damage. This system was able to extract actionable information from the crisis related twitter data. In [7] the authors classify tweets into events based classes. They use a convolutional neural network based model for classifying earthquake-related tweets into informative or non-informative classes.

Inspired from the related works, the need for an automated disaster response system based on the tweets about the disasters in India is proposed through the Disastro framework. The main objectives of Disastro are threefold: 1) build an automated system that provides relevant tweets related to rescue and donation to the rescue volunteers 2) investigate the various machine learning algorithms towards building an efficient classification model. 3) understand the importance of representation of text through various vectorization techniques.

II. DATA COLLECTION

The dataset used for developing Disastro comprises of tweets tweeted during the particular disaster. The tweets for the dataset were collected using Web Crawler, an Internet bot which helps in Web indexing. Around 23,000 tweets were extracted using Python WebCrawler through the utilization of various hashtags that were trending during the disaster. The detail of the extracted dataset is as shown in Table 1.

TABLE I. NUMBER OF TWEETS COLLECTED AND THEIR CORRESPONDING HASHTAGS.

Disaster	Hashtags used	Number of tweets
Chennai Rains 2015	#ChennaiRains, #ChennaiRainsHelp, #ChennaiFloods, #ChennaiVolunteer,#ChennaiRescue, #GajaCyclone	15,980
Kerala floods 2018	#KeralaFloods2018, #KeralaFloods, #OpMadad	7000

978-1-7281-4894-6/19/$31.00 ©2019 IEEE
DOI 10.1109/HiPCW.2019.00029

III. Data Pre-Processing

Data pre-processing plays a vital role in text classification. NLTK library and Regex in Python are used for the Data Pre-processing tasks. Some of the pre-processing tasks include

- Removal of tweets which are not posted in English.
- Removal of tweets containing less than five words.
- Conversion of all words to lowercase.
- Removal of hashtags, emoticons and alpha numeric characters.
- Removal of the RT's and CC's used in the tweets.
- Removal of the URL links, punctuations and extra white spaces used in the tweets.
- Replacement of phone numbers with a special word("#######MOBNO#######").
- Reduction of the words to their lemmatized form.

IV. Proposed Methodology

A. Word Representation

It is the basic building block of data representation that plays a crucial role in determining the performance of the model. Some of them are,

1) Count Vectorizer

It works on Terms Frequency, i.e. counting the occurrences of tokens and building a sparse matrix. The size of the vector is always proportional to the size of our vocabulary. This representation does not preserve the order of the words in the original sentences.

2) TF-IDF

TF-IDF stands for term frequency-inverse document frequency. TF-IDF weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Term Frequency (TF) is a scoring of the frequency of the word in the current document. Since every document is different in length, it is possible that a term would appear much more time in long documents than shorter ones. The term frequency is often divided by the document length to normalize.

$$TF(X) = \frac{Number\ of\ times\ a\ word\ 'X'\ has\ occurred\ in\ the\ document}{Total\ Number\ of\ words\ in\ the\ document} \quad (1)$$

Inverse Document Frequency (IDF)is a scoring of how rare the word is across documents. IDF is a measure of how rare a term is. Rarer the term more is the IDF score.

$$IDF(X) = \log\left(\frac{Total\ Number\ of\ Documents}{Number\ of\ document(s)\ that\ contain(s)\ the\ words\ "X"}\right) \quad (2)$$

$$TF\text{-}IDF(X) = TF(X) * IDF(X) \quad (3)$$

Bi-Gram TF-IDF is used for the representation of the words.

B. Two level Classification

The design of Disastro is a two step classification process. The first step filters the disaster relevant tweets, from the extraneous ones. The second step takes the output of the first step and classifies the tweets into 'donation' or 'rescue' class. This enables the disaster relief volunteers to make a choice based on the type of service they wish to render – rescue or donation. The design of the classifier is shown in Figure 1.

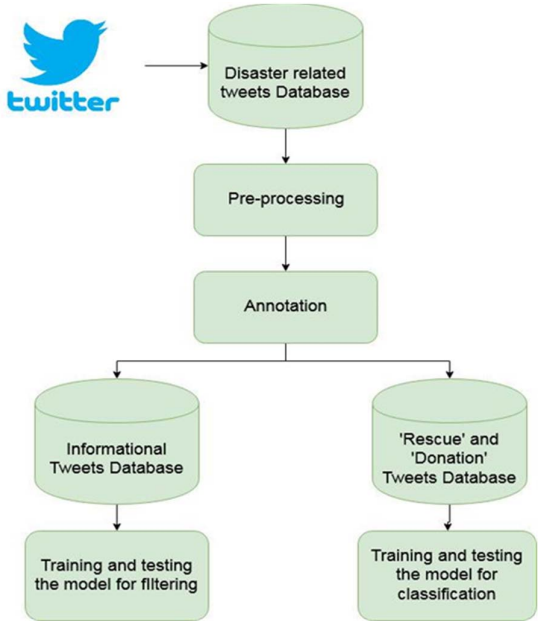

Figure. 1 Design methodology of the two step classifier

1) Filter

The objective of the filter is to differentiate and discard the irrelevant tweets from the tweets that were collected at the time of disaster. The tweets which contain potential information regarding a need for resource or help are considered to be informational tweets. A binary class or multi-class approach to this filtering problem is not suitable since the data has only one class. One class SVM is trained on data that has only one class, which is the "normal" class. The tweets classified under the "normal class" are considered to be informational tweets. Other tweets are considered as non-informational and are discarded. One class SVM is compared over both Count Vectorizer and TF-IDF vectorization techniques for understanding the proficient way of word representation. One class SVM with bigram TF-IDF representation yielded better results compared to One class SVM with Count Vectorizer representation with an accuracy of above 82%. The results are as shown in Table II.

TABLE II. PERFORMANCE OF ONE-CLASS SVM FOR VARIOUS VECTORIZATION TECHNIQUES.

Word to Vector representation	Accuracy	F1 score	Precision	Recall
CountVectorizer	0.4600	0.1390	0.070	0.770
TF-IDF	0.8297	0.8687	0.8098	0.9376

2) Classifier

At the second stage, the informational tweets are classified into either of the following class

- Donation: Tweets that are related to appeal for food supplies, medicines, blankets etc.
- Rescue: Tweets that are related to appeal for rescue missions to save lives from crisis.

Five different machine learning algorithms were used for building the classifier. They are,

- *Logistic Regression:* Logistic regression [14] is a linear regression technique that can be used to predict binary-class instances. The algorithm uses LogitBoost [13] to build the regression model and was further improved by Sumner et al. [15] to increase the speed of the model construction.
- *Naive Bayes classifier:* is a simple but powerful learning algorithm based on Bayes' theorem [12]. During the training phase the dataset is analyzed and a probabilistic model is built based on the learned attributes. The model assumes that all the attributes are independent to each other. This assumption is called class conditional independence. The heart of the classifier is based on the Bayes theorem.
- *Stochastic gradient descent:* Regularised linear classification models with Stochastic Gradient Descent is implemented by SGD classifier[11]. The gradient of the loss is estimated in each sample at a time and the model is updated along the way with a decreasing strength schedule. SGD allows mini batch learning unlike gradient descent which uses the entire dataset for each update.
- *Support vector classifier:* Support vector machine proposed by Cortes.et al[10] is a two-group classifier that classifies the two classes by separating them using a hyperplane. The input vectors are non-linearly mapped to a higher dimensional feature space and a linear surface is constructed in the high dimensional feature space. For multi-class classification the algorithm separates the classes into two main classes. Each class is further separated until the target class is obtained.
- *Random Forest:* A random forest algorithm generates random forest which is a group of decision trees[9]. As more trees are generated each tree predicts a class. The class that is predicted by more number of trees is the result of classification. The strength of random forest lies in the fact that a large number of trees as a collection performs well than individual trees.

C. Implementation and Results

The above discussed machine learning algorithms for classification was implemented on Intel i7-7700HQ @ 2.80GHz with 8 GB RAM. The average time for training One-class SVM and logistic regression was around 299 seconds and 0.5 seconds respectively. The performance of the machine learning algorithms to model the application was evaluated via Accuracy, Precision, Recall & F1 Score metrics. Table III shows the comparison of the algorithms for the different performance metrics. The data collected for developing this model is linearly separable. Also the model produces a binary classification with the two classes 'Rescue' and 'Donation'. Hence, we can observe that, Logistic Regression outperforms the other classifiers

TABLE III. PERFORMANCE COMPARISON OF VARIOUS MACHINE LEARNING ALGORITHMS IN CLASSIFICATION.

Algorithm	Accuracy	F1 Score	Precision	Recall
Logistic Regression	0.9407	0.9415	0.9602	0.9235
Naive - Bayes Classifier	0.8970	0.8987	0.9188	0.8794
SGD	0.9390	0.9391	0.9459	0.9325
Random Forests	0.9150	0.9141	0.9087	0.9195
Linear SVC	0.9234	0.9230	0.9230	0.9230

The 8500 training samples to One class SVM contained 3196 disaster related tweets and 5304 non disaster tweets. The confusion matrix for the classification output One class SVM is as shown in Figure. 2.

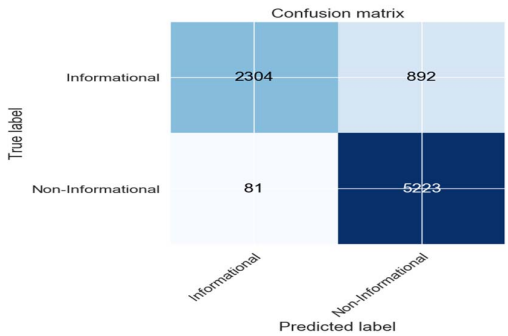

Figure. 2 Confusion Matrix for One class SVM output

Similarly, the 2289 training samples to Logistic Regression contained 1106 'Donation' class tweets and 1183'Rescue' class tweets. The confusion matrix for the classification output Logistic Regression is as shown in Figure. 3.

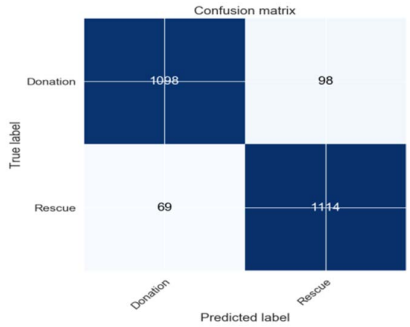

Figure. 3 Confusion Matrix for Logistic Regression output

D. Disastro: Enduser Perspective

The disaster relief volunteers are the end users of the Disastro framework. The end users of Disastro invoke the server for information through a data fetch request that gets their name, e-mail id, location and type of service they wish to provide (either rescue or donation). Figure. 4 shows the user interface of web page to fetch the data by the end user. The high level flow diagram of the Disastro framework from the end user perspective is shown in Figure. 5.

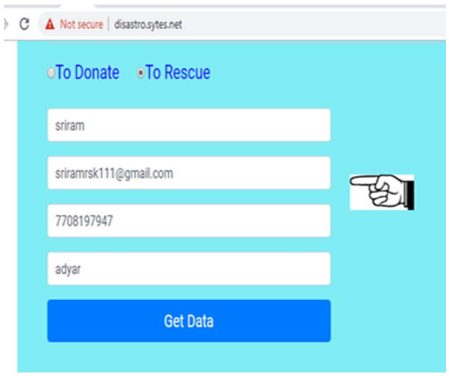

Figure. 4 Snapshot of data fetch request through the web page

Figure. 6 shows the grouped relevant tweets based on the choice made in Figure. 4 as Rescue.

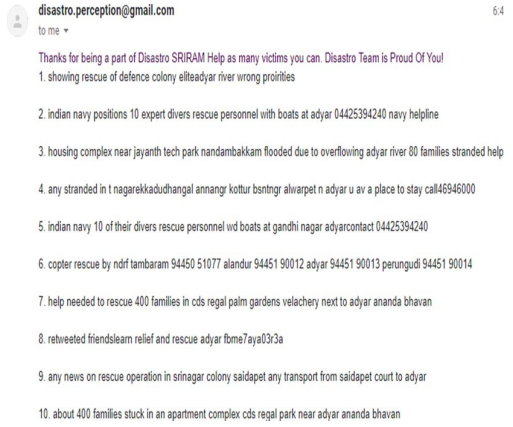

Figure. 6 Response sent by the system for the query

Disaster response system developed for a particular disaster significantly drops in classification accuracy when used for different but similar disasters [8]. The robustness of Disastro lies in its ability to provide reasonably good performance across similar disasters without much deviation in its classification accuracy. This is verified across two Indian flood-related disasters as shown in Table IV.

TABLE IV. PERFORMANCE COMPARISON OF DISASTRO ACROSS DIFFERENT DISASTERS.

Types of Disaster	Accuracy
Chennai Rains	0.9407
Kerala floods	0.9289

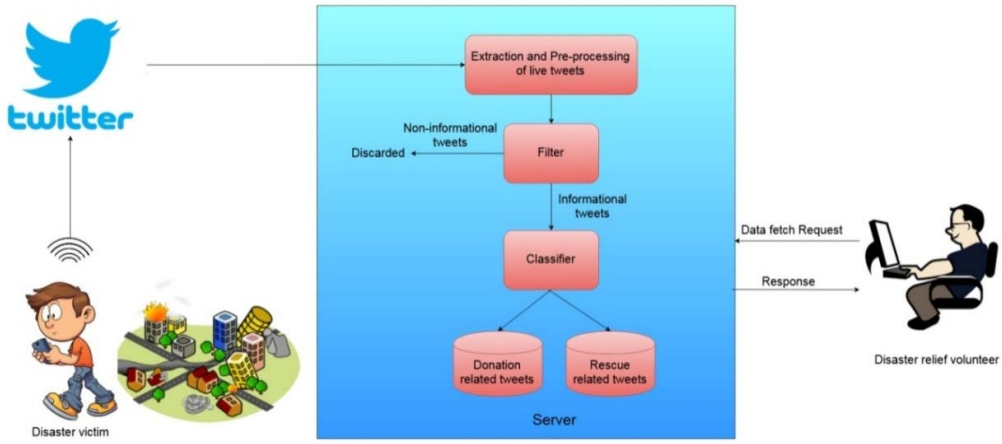

Figure. 5 Disastro Framework

V. CONCLUSION

A micro-blogging network such as twitter has an overwhelming data flow at all times. During disasters twitter contains enormous amount of informations related to the victim's needs. Retrieval of relevant information from them is a challenging task. Identifying informational tweets and classifying them under "rescue" and "donation" domains could felicitate the disaster volunteers in identifying the appropriate victims as per the type of service they wish to render and impart it efficiently. Disastro the automated disaster response system solves this need of the disaster relief volunteers through the aid of artificial intelligence. The logistic regression classifier provides an accuracy of around 94% compared to the other machine learning algorithms such as Naive – Bayes, SGD, Random Forests and Linear SVC. The improved classification accuracy and efficiency of feature representation utilized for the application development proves its excellence for real time disaster response. The versatility of Disastro across locations makes it proficient and reliable for resilience. In future, we would consider increasing the number of classes. Each class would represent tweets more precise to the type of trap from which the victim needs to be rescued and the type of resource the victim need as donation. This would render the explicit need of the victims and ease the rescue volunteers.

References

[1] Kumar, S., Morstatter, F., & Liu, H. (2014). *Twitter data analytics* (pp. 1041-4347). NewYork:Springer.

[2] Steiger, E., De Albuquerque, J. P., & Zipf, A. (2015). An Advanced Systematic Literature Review on Spatiotemporal Analyses of T witter Data. *Transactions in GIS*, *19*(6), 809-834.

[3] Williams, S. A., Terras, M. M., & Warwick, C. (2013). What do people study when they study Twitter? Classifying Twitter related academic papers. *Journal of Documentation*, *69*(3), 384-410.

[4] Laylavi, F. (2016). *A framework for adopting Twitter data in emergency response* (Doctoral dissertation).

[5] Imran, M., Castillo, C., Lucas, J., Meier, P., & Vieweg, S. (2014, April). AIDR: Artificial intelligence for disaster response. In *Proceedings of the 23rd International Conference on World Wide Web* (pp. 159-162). ACM.

[6] Ashktorab, Z., Brown, C., Nandi, M., & Culotta, A. (2014, May). Tweedr: Mining twitter to inform disaster response. In *ISCRAM*.

[7] Nguyen, D. T., Al Mannai, K. A., Joty, S., Sajjad, H., Imran, M., & Mitra, P. (2017, May). Robust classification of crisis-related data on social networks using convolutional neural networks. In *Eleventh International AAAI Conference on Web and Social Media*.

[8] Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., & Meier, P. (2013, May). Practical extraction of disaster-relevant information from social media. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 1021-1024). ACM.

[9] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

[10] C. Cortes and V. Vapnik. Support-vector networks. Machine Learning, 20:273–297, 1995.

[11] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Y. Lechevallier and G. Saporta, editors, Proc. 19th Int'l Conf. on Computational Statistics (COMPSTAT'2010), pages 177–187, Paris, France, August 2010. Springer.

[12] D. Lowd and P. Domingos. Naive bayes models for probability estimation. In Proc. 22nd Int'l Conf. on Machine Learning (ICML '05), pages 529–536. ACM, 2005.

[13] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. The annals of statistics, 28(2):337–407, 2000.

[14] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. 95(1-2):161–205, 2005.

[15] M. Sumner, E. Frank, and M. Hall. Speeding up logistic model tree induction. In 9th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD '05), pages 675–683. Springer, 2005.