Evaluation of rule-based, CountVectorizer, and Word2Vec machine learning models for tweet analysis to improve disaster relief

Radhika Goyal
Lynbrook High School
San Jose, California, USA
radhikagoyal3000@gmail.com

2021 IEEE Global Humanitarian Technology Conference (GHTC) | 978-1-6654-3372-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/GHTC53159.2021.9612486

Abstract*—Social media platforms, such as Twitter, are being increasingly used by people as a means of requesting help during disaster events. Machine learning techniques can be used to identify tweets containing requests for help and classify them based on the type of aid that is being requested. In this paper, we build and compare three different models to classify tweets requesting aid into the categories "food," "water," "energy," and "medical" by using the tweet dataset from Hurricane Sandy, which took place in 2012. Our first model uses a rule-based classifier. Our second model is based on the Scikit-learn toolkit's CountVectorizer, and the third model uses the Word2Vec based classifier. We found that the machine learning models based on CountVectorizer and Word2Vec have higher accuracy than the rule-based classifier model. We also show that the model based on Word2Vec provides the highest accuracy among the three models.

Keywords*—Disaster management, Hurricane Sandy, Social media, Twitter, Machine Learning

I. INTRODUCTION

When natural disasters such as hurricanes strike, providing relief aid to the victims in a timely manner is crucial. With the ubiquitous use of smartphones, it is becoming more common for the victims to post information on social media platforms such as Twitter about the specific location where they are stuck and the type of aid they are seeking, both during and after a crisis event [1]. Twitter served as a highly valuable source of disaster-related information during Hurricane Sandy in 2012 [2]. Research on using social media as a data source to understand various disasters [3] has been growing, with applications such as situational awareness [4] [5] and understanding public sentiment [6]. Previous work has had success in applying machine-learning classifiers to detect tweets in broad categories, such as disaster type and relevance. This work focuses on classifying tweets based on the type of resource being requested. If the data about the specific type of aid needed at a specific location can be extracted from the tweets accurately and promptly, disaster relief organizations and volunteers can use this data to help fulfill these needs during disaster events [7][8].

Manual ways of looking through a substantial number of tweets to find the tweets requesting aid can be cumbersome and time-consuming. Automatic classification of tweets to identify useful resource-request related tweets is a much needed but challenging task. As such, Twitter data presents several challenges to the traditional natural language processing (NLP) methods because twitter messages are very short, with a character limit of 140 characters, and often lack proper written grammar and contain abbreviations or informal language such as "omg" for "Oh my God." Furthermore, there is no consistent format that clearly suggests whether someone is requesting or donating an item. These issues make traditional NLP tools such as parsers inadequate for processing such data.

Another challenge is that resource-related tweets account for only a small portion of the large number of tweets generated during major crisis events. In the tweet dataset from Hurricane Sandy used in this paper, only about five percent of the tweets are resource-related requests or offers [8]. Although resource-related tweets are usually a small percentage of the total number of tweets during a crisis event, the problem of identifying such requests is worth solving because fulfilling even a few requests can help save lives. An example of one such tweet message is "I need medical help: please call an ambulance."

II. LITERATURE REVIEW

Extensive research has been done to classify disaster related twitter data. Most of the previous work has focused on building classifiers to identify whether tweets are about a disaster event or not [9], the type of event (floods, hurricane, etc.) [10], the type of information (affected individuals, damaged infrastructures, donations and volunteer, etc.) [9], [10], and whether the request for help is urgent or not [11].

Typically, there are two types of approaches used to build tweet classifiers: keyword matching-based approach [12], [13] and machine learning-based approach [10], [11]. The authors in [12] use keyword and hashtag matching to identify tweets relevant to a disaster and compare this approach to a learning-based system. The work in [10] uses different machine learning models such as random forest, naïve Bayes, and Support Vector Machine models to classify tweets from 19 different natural disasters e.g., floods and earthquakes into categories such as injured or dead, sympathy and emotional support, etc. The work in [11] compares different machine learning models to identify the urgent requests from the tweets.

When building machine learning-based models for tweet classification, word embeddings are often used to convert the words (text data) in the tweets into word vectors (numerical data) because machine learning classifiers expect data in numerical form. The work in [14] uses two types of word embeddings: CountVectorizer [15] and TfidfVectorizer [16], and compares their performance when used with different machine learning classifiers in identifying disaster-related tweets. The authors conclude that methods using CountVectorizer provide higher performance than methods using TfidfVectorizer. Their work focuses on classifying tweets based on whether or not the tweets are disaster-related. Some work [8] has been done to identify resource needs and availabilities by applying regular expressions and TfidfVectorizer.

The focus of our work is to build and compare rule-based (also known as keyword matching-based) classifier and machine learning-based classifiers that use word embeddings to specifically classify resource request related disaster tweets into these four fine-grained categories – "food," "water," "energy," and "medical". All other tweets that do not belong to any of these four categories are put into the "other" category. Also, the classification of tweets based on whether a tweet is a request for a resource or an offer for help is not the focus of this work. Our goal is to find the most accurate model between the three proposed models.

III. DATA AND DATA PRE-PROCESSING

A. Dataset

We use the already labeled tweet dataset of 2012 Hurricane Sandy [8]. It consists of two separate corpora of tweets: one containing resource requests, and the other containing resource offers. We only used the corpus that consists of the resource requests.

B. Data Preprocessing

In this sub-section, we describe the data pre-processing steps that are applied in order to clean the tweet data before it can be used in the second and third models, both of which are machine learning-based models. We use the following steps to preprocess the data:

- First, we normalize the data by converting the characters to lower-case.
- Next, we remove unwanted punctuation marks as comma, colons, semicolons, as they don't provide any valuable information.
- Then, we split the sentences in the tweets into individual words or tokens. This process is called tokenization.
- Further, we apply stemming and lemmatization methods to convert each word to its common base form. Stemming looks at the current word only, while lemmatization also takes the context into consideration. For example, after stemming, "walking" is replaced with "walk." After lemmatizing, "better" is replaced with "good." We use the powerful Natural Language Toolkit (NLTK) for both stemming and lemmatization.
- As a final data preprocessing step, we remove the stop words from the tweets. Stop words are words that occur in all the categories and are not relevant to the context, such as 'at', 'is', 'the', and so on. It is usually advantageous for the classifier to ignore these stop words since they may add noises or cause numerical issues as they add baggage to the model.

Below is an example tweet before and after the above-mentioned data pre-processing steps:

Original tweet:
need flashlights at 2374 38th street astoria blvd
Cleaned and tokenized tweet data:
['need', 'flashlights', '2374', '38th', 'street', 'astoria', 'blvd']

C. Data splitting

We split the data into training data and test data. Eighty percent of the data is used to train the models while the remaining twenty percent of the data is used to test the models.

IV. METHODOLOGY

A. First Model: Rule-based classifier

Rule-based classifier uses explicitly written rules to match keywords in order to determine the category that a tweet belongs to [12], [13]. This model assumes that we have some idea from previous disaster events of the types of keywords to look for based on which the category is decided. For example, if a tweet contains the word "flashlight" or "batteries," the classifier would classify it under the "energy" category. If a tweet contains words such as "hungry" or "starving," it would fall under the "food" category. We use the following steps to build this classifier:

- First, we normalize the data by converting the characters to lower-case.
- Next, we create this model by writing explicit rules using conditional statements in the Python programming language.
- Finally, we run all the tweets in the dataset and compare the categories predicted by the model with the actual categories mentioned in the labeled dataset.

B. Second Model: CountVectorizer combined with Logistic regression-based classifier

First, we pre-process the data by applying the steps already mentioned in the data pre-processing sub-section of Section III. The resulting data, which consists of words, still cannot be input directly to machine learning based models because these classifiers expect input in the form of numerical feature vectors of fixed size. Therefore, we use CountVectorizer from the Scikit-learn toolkit [15] to convert the word data in each of the tweets into a corresponding fixed size numerical data.

CountVectorizer is a method that converts a document (which, in our case, is the tweet data) into vectors by counting the number of occurrences of each word in that document.

CountVectorizer has 2 methods – fit and transform. The fit method transforms the training data to vectors and trains the

logistic regression classifier. The transform method just transforms the data into matrix, there is no learning here.
Steps:

- First, we pre-process the data.
- Next, we split 80% of the tweets as training data and 20% as test data.
- We call the CountVectorizer's fit method on the training tweet data to train the logistic regression model classifier.
- Finally, we call the CountVectorizer's transform method on the test data to predict the category of each of the tweet in the test data and compare it with the true labels to compute the accuracy.

C. Third Model: Word2Vec combined with Logistic regression-based classifier

Our third model is a machine learning model that uses the Word2Vec tool [17] to map words to word vectors.

Word2Vec is a method that converts each unique word into its own multi-dimensional vector so that words with similar meanings have similar word vectors. In our work, each vector consists of 300 numbers, which is hard to interpret by looking at the numbers.

Word similarity is measured by the distance between the endpoints of two word-vectors. A metric called cosine similarity is used to compare the word vectors of two words. When two words are similar, the cosine similarity score of their word vectors is higher, and when two words are dissimilar, the cosine similarity score is lower. For example, the words "good" and "great" have similar word vectors, and the words "good" and "earth" have different word vectors. Using this approach, it can be determined whether a tweet should belong to the same category as other similar tweets.
Steps:

- First, we pre-process the data by applying the steps already mentioned in the data pre-processing sub-section of Section III.
- Next, we split 80% of the tweets as training data and 20% as test data.
- Then we use Word2Vec method on the cleaned and tokenized tweets in the training set as well as the test set. Given word vectors of each word in a tweet, we then compute the vectors for each tweet by finding the average of the word vectors for all the words in the tweet.
- Next, we train the logistic regression-based classifier using the vectors of the training dataset.
- Finally, use the vectors of the test data to make predictions about the category of tweets in the test data.

V. RESULTS

We compute the accuracy of each of the three models. The accuracy of a model is measured by counting the number of correct classifications and dividing that by the number of attempted classifications. For each model, we first measure the accuracy for each of the four individual categories – "food," "water," "energy," and "medical". Then, we measure the overall accuracy of the model by counting the total number of correctly classified tweets across all of these four categories and dividing it by the total number of attempted classifications across these four categories.

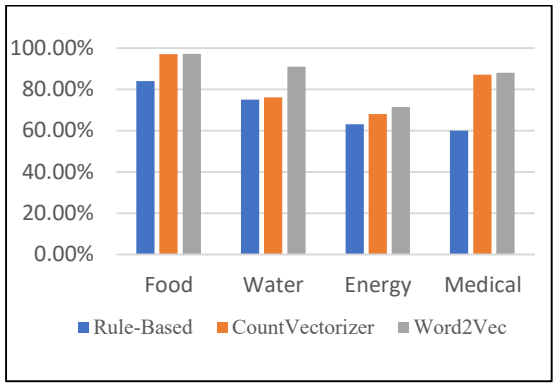

Figure 1: Per-category accuracy for each model

TABLE 1: Per category accuracy for each model

Category/ Accuracy	Rule Based	Count Vectorizer	Word2Vec
Food	84.00%	96.99%	97.08%
Water	75.00%	76.00%	90.90%
Energy	63.00%	68.00%	71.40%
Medical	60.00%	87.00%	88.00%

TABLE 2: Overall accuracy for each model

Category	Overall Accuracy
Rule-Based	70%
CountVectorizer	82%
Word2Vec	86%

Figure 1 and Table 1 show the per-category accuracies for each model. The per-category accuracies of the rule-based classifier for the four categories – "food," "water," "energy," and "medical" are 84%, 75%, 63%, and 60% respectively. The per-category accuracies of the CountVectorizer and logistic regression-based model for the four categories – "food," "water," "energy," and "medical" are 96.99%, 76%, 68%, and 87% respectively. The per-category accuracies of the Word2Vec and logistic regression-based model for the four categories – "food," "water," "energy," and "medical" are 97.08%, 90.90%, 71.40%, and 88% respectively.

Table 2 shows that the rule-based classifier model is the least accurate with an overall accuracy of 70%, CountVectorizer model has an accuracy of 82%, and the Word2Vec model has the highest accuracy of 86%.

Limitations of the models:

A major limitation of the rule-based classifier is that it uses pattern-matching regardless of context. For example, if

the tweet consists of the word "food" out of context such as this tweet – "*Waiting out #Sandy by reading Plato. Food for thought,"* the rule-based system looks for the word "food" and incorrectly classifies it under the category "food" even though this tweet is not requesting food.

A limitation of the CountVectorizer approach is that it only looks at the counts of words in each tweet and does not take into account the meaning of the words.

Word2Vec approach takes into account the meaning as well as the semantics of the words.

VI. CONCLUSION

A. Contributions

Our work is focused on building and comparing three different approaches to find the most promising approach for classifying resource-related tweets during a disaster event. Firstly, we built a rule-based classifier to classify tweets. Secondly, we implemented tweet classification using CountVectorizer and Word2Vec features in order to determine which of these word vectorizers when combined with a logistic regression classifier would provide more accurate tweet classification of resource requests.

B. Main findings

Our results show that the machine learning-based classifiers and models outperform the rule-based classifiers. We find that the best-performing classifier is the logistic regression model trained on Word2Vec embeddings. This model achieves an accuracy of over 0.86, confirming that a system based on it can indeed be used in facilitating disaster management by detecting tweets containing useful resource-related information. Although Word2Vec is the most promising, CountVectorizer is also quite close with an accuracy of 0.82. The rule-based model, however, would be inefficient as it compromises the accuracy of the tweet classification process.

C. Future work

The machine learning-based models presented in this paper can be enhanced by adding more complexity, such as by adding more layers to improve their accuracy on this task. These models can also be improved by further training and testing them with datasets from multiple hurricane events.

In this work, we only focused on hurricane data. For future work, similar models can be built for other types of natural hazards such as tornadoes, earthquakes, floods, or wildfires by training and testing the models using the twitter data generated during these disaster events.

REFERENCES

[1] Landwehr, P. M., & Carley, K. M. (2014). Social media in disaster relief: Usage patterns, data mining tools, and current research directions. In W. C. Wesley (Ed.), Data mining and knowledge discovery for big data studies (pp. 225–257). Heidelberg, New York, Dordrecht, London: Springer.

[2] N. Pourebrahim, S. Sultana, J. Edwards, A. Gochanour, S. Mohanty, Understanding communication dynamics on Twitter during natural disasters: A case study of Hurricane Sandy, Int. J. Disaster Risk Reduct. 37 (2019) 101176.

[3] Jie Yin, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power. 2012. Using social media to enhance emergency situation awareness. IEEE Intelligent Systems, 27(6):52–59.

[4] Sarah Vieweg, Amanda L. Hughes, Kate Starbird, and Leysia Palen. 2010. Microblogging during two natural hazards events: what Twitter may contribute to situational awareness. In CHI.

[5] Sudha Verma, Sarah Vieweg, William J Corvey, Leysia Palen, James H Martin, Martha Palmer, Aaron.
Schram, and Kenneth Mark Anderson. 2011. Natural language processing to the rescue? extracting" situational awareness" tweets during mass emergency. In ICWSM.

[6] Son Doan, Bao Khanh Ho Vo, and Nigel Collier. 2012. An analysis of Twitter messages in the 2011 Tohoku earthquake. In Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, volume 91 LNICST, pages 58–66.

[7] Kaufhold, M. A., & Reuter, C. (2016). The self-organization of digital volunteers across social media: The case of the 2013 European floods in Germany. *Journal of Homeland Security and Emergency Management*, 13(1), 137– 166.

[8] Purohit, H., Castillo, C., Diaz, F., Sheth, A., & Meier, P. (2013). Emergency-relief coordination on social media: Automatically matching resource requests and offers. *First Monday*, *19*(1). [Online]. Available: https://doi.org/10.5210/fm.v19i1.4848 [Accessed: 12-Feb-2021]

[9] G. Burel, H. Saif, M. Fernandez, H. Alani, On Semantics and Deep Learning for Event Detection in Crisis Situations, in: Workshop on Semantic Deep Learning (SemDeep), at the European Semantic Web Conference (ESWC) 2017, Elsevier, 2017. http://oro.open.ac.uk/49639/.

[10] M. Imran, P. Mitra, C. Castillo, Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), European Language Resources Association (ELRA), Portoroˇz, Slovenia, 2016.

[11] Devaraj, Ashwin, et al. "Machine-Learning Methods for Identifying Social Media-Based Requests for Urgent Help during Hurricanes." *International Journal of Disaster Risk Reduction*, Elsevier, 20 July 2020.

[12] H. To, S. Agrawal, S. H. Kim, and C. Shahabi, "On identifying disaster related tweets: Matching-based or learning-based?." In 2017 IEEE Third International Conference on Multimedia Big Data (BigMM) (pp. 330-337), April 2017.

[13] Z. Ashktorab, C. Brown, M. Nandi, and A. Culotta. "Tweedr: Mining twitter to inform disaster response." In Proceedings of the 11th International ISCRAM Conference, May 2014.

[14] N. Assery, Y. Xiaohong, S. Almalki, R. Kaushik and Q. Xiuli, "Comparing Learning-Based Methods for Identifying Disaster-Related Tweets," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019, pp. 1829-1836, doi: 10.1109/ICMLA.2019.00295.

[15] F. Pedregosa, et al. "Sklearn.feature_extraction.text.CountVectorizer." 2013.

[16] F. Pedregosa, et al. "sklearn.feature_extraction.text.TfidfVectorizer." 2013.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in Neural Information Processing Systems, 2013, pp. 3111–31