NLP with Disaster Tweets

Bopparthi, Sruthi Sridhar*,

[1]*School of Information Technology, University of Cincinnati, Ohio, USA*

Email Address : bopparsr@mail.uc.edu

Abstract

People mainly use human language to talk to each other, but it's quite complicated, unclear, and can change a lot. Recently, new algorithms for understanding language better have gotten a lot of attention from researchers. When there's a disaster, people share updates on social media like Twitter. This information is super important for teams helping with the disaster because it tells them what's happening right away. Text mining and machine learning can go through all the big piles of information on social media, like Twitter, and find important details by looking for specific words related to disasters. But there's a problem – sometimes, the words might be used in a different way, like as a metaphor, and that can make the computer get things wrong. So, this research is about using Natural Language Processing (NLP) and special computer programs to figure out if a tweet is really about a disaster or if it's using those words in a different way. This study presents a comparison between four different models and then selecting the best model to act upon. I compared the accuracy of all four models, Logistic regression was standing out in all of the models with an accuracy of around 80% with Precision of 79.90%, Recall of 88% and with F1 Score of 83%. This model performed the best in all four models.

Keywords: Disaster, Natural Language Processing, Machine Learning, Twitter, Tweets, Text Analysis

1. Introduction

Twitter started in 2006 as a small messaging service, but over the years, it has evolved into a microblogging and social networking platform known for its short messages called "tweets." Users can post, like, and retweet tweets, creating a vast amount of unstructured data. This data includes various content, from essays to poems, making it a unique challenge for analysis(Berenice & Pedro).

As of October 2019, around 6000 tweets are generated every second globally, totaling 500 million tweets per day or 200 billion tweets per year. This immense volume of data, produced by millions of users worldwide, poses a challenge due to its unstructured nature. Natural Language Processing (NLP) is crucial for understanding and analyzing this diverse set of information.

In the context of emergencies and disasters, Twitter has become a vital communication channel. The widespread use of smartphones and other devices enables people to share real-time information about ongoing disasters. Analyzing Twitter data has become important for various data-analytics agencies, news organizations, and disaster relief groups. The ability to monitor and analyze tweets in real time allows for the identification of disaster occurrences, potentially helping millions of people take evasive actions and aiding government agencies in timely responses and evacuations (Humaid Alhammadi).

This research specifically focuses on analyzing 7613 tweets, each labeled with a tweet ID, location, tweet keyword, and class (either "Disaster-related" or "Not Disaster-related"). The goal is to train a classifier model, using the different algorithms, to predict the class of tweets. The model's accuracy is evaluated using a Confusion Matrix, and the AUC score is calculated as a measure of accuracy. The research question here is to classify tweets accurately, distinguishing those related to disasters and which are non disasters but tweeted metaphorically as disasters.

This study had limitations due to the availability of the dataset. Disaster-related agencies often have confidentiality restrictions, preventing the sharing of real-time information on how specific text can be flagged. Consequently, an offline dataset collected from Twitter was utilized, consisting of thousands of tweets labeled as either real or fake disaster-related tweets. The study is constrained by this dataset, which may only capture a portion of disaster-related tweets, and it is based on offline rather than real-time tweets.

The rest of this study is organized as follows. In Section 2, will delve into the methodology employed to develop the proposed decision-fusion-based SMS spam detection models. Section 3 will provide a detailed explanation of the methodology used for training different models. Study's findings and performance comparisons with well-known models will be summarized in Section 4. Finally, will conclude the study and outline future works in Section 5.

2. Methodology

In crafting this study, I designed a framework represented by Figure 1, delineated into three key phases. Phase I involves activities such as data download, pre-processing, data visualization, and partitioning. Shifting to Phase II, my focus turns to data modeling, employing Multiple Algorithms. Concluding in Phase III, my attention centers on evaluating the performance of the models. Further insights into each phase will be provided in the subsequent sections.

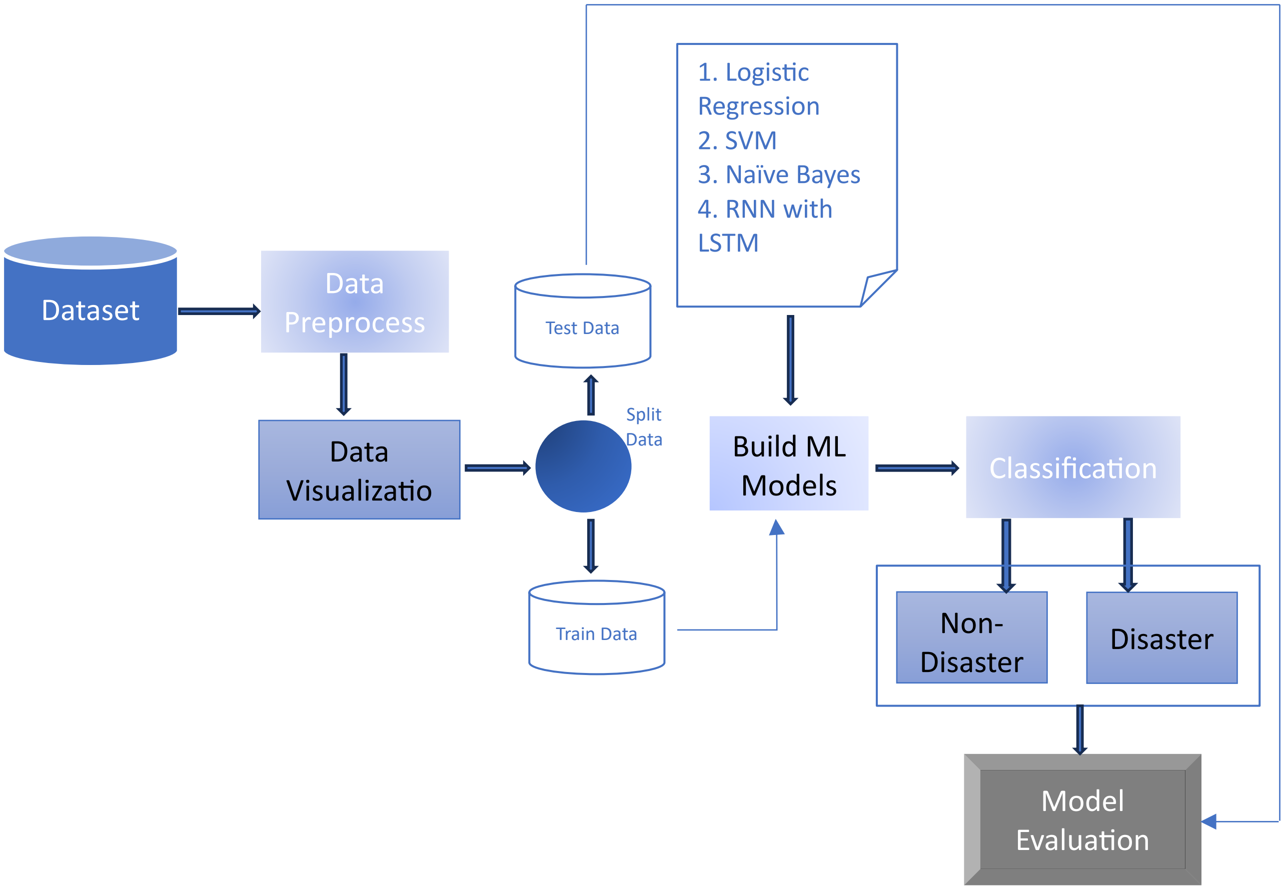


Figure 1: Methodology

2.1 Phase I: Study Data

This paper utilized a dataset sourced from Kaggle, credited to (Addison, Devrishi, Phil, and Yufeng). Given the vast number of daily tweets on the Twitter website, the dataset chosen for this study encompassed 7040 tweets, subsequently divided into training and testing data. The training dataset featured four columns: Keyword, Location, Text, and Target. Meanwhile, the testing dataset included three columns: Keyword, Location, and Text.

The initial steps involved downloading the dataset from Kaggle to the local computer and loading it into Jupyter notebook for data modeling. An assessment for missing or null values revealed 61 null values in the Keyword column and 2533 null values in the Location column. These were addressed by filling in the null values.

Addressing irregularities in the Text column, processes included the removal of URLs, HTML tags, emojis, and punctuation. Tokenization followed, involving the deletion of stop words or symbols like 'the', '.', '*', and '/'. This step aimed to eliminate infrequently used words and symbols in tweets, reducing unnecessary word characteristics. Attempts to detect outliers using the Z-Score method indicated the absence of outliers in the data.

Moving forward, the study delved into visualization. The top 10 locations were displayed, categorized by disaster and non-disaster labels. Major keywords for disaster and non-disaster tweets were showcased, along with the top words from negative and positive tweets.

2.2 Phase II: Decision Fusion Model

In many instances, different models yield diverse classifications for tweets. Instead of relying on a single model in isolation, combining multiple models often results in improved tweet classification. Fusion involves bringing together data or information from various sources, and within this hierarchy, there are three degrees: data fusion, feature fusion, and decision fusion. Decision fusion, specifically, involves amalgamating predictions from different base learners.

After preprocessing the data, the dataset was split into 80:20 training and testing data respectively, so that we can train the model with 80% of data and test the model with 20% of data.

The core concept is to base the final detection on the collective knowledge of the models about the situation. This paper embraced four decision fusion models with proven success in similar work across different fields.

- **SVM** : Support Vector Machine (SVM) is a machine learning algorithm used for classification and regression tasks. It works by finding a hyperplane that best separates the data points into different classes. SVM aims to maximize the margin, which is the distance between the hyperplane and the nearest data points from each class. This algorithm is effective in high-dimensional spaces and is widely used in various applications, including text classification, image recognition, and bioinformatics.
- **Logistic Regression** : Logistic Regression is a statistical model used for binary classification tasks, where the outcome is a binary variable (usually labeled as 0 or 1). Despite its name, logistic regression is used for classification, not regression. It models the probability that an instance belongs to a particular class and predicts the probability scores between 0 and 1. The logistic function (sigmoid function) is employed to map the output of the linear

combination of input features into the probability space. Logistic Regression is commonly used in fields such as medicine, economics, and social sciences for predicting categorical outcomes.

- **Naive Bayes** : Naive Bayes is a probabilistic machine learning algorithm based on Bayes' theorem. Despite its "naive" assumption of independence between features, it performs well in various classification tasks. It's particularly popular for text classification, spam filtering, and sentiment analysis. The algorithm calculates the probability of each class given a set of features and selects the class with the highest probability as the predicted class. Despite its simplicity, Naive Bayes can be effective and computationally efficient, making it a suitable choice for certain applications.
- **RNN with LSTM** : Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) units are a type of neural network architecture designed for sequence data. LSTMs address the vanishing gradient problem associated with traditional RNNs, allowing them to capture long-range dependencies in sequential data.
- In an RNN with LSTM, each LSTM unit maintains a memory cell that can store information over long periods. This enables the network to remember relevant information from earlier parts of the sequence and use it in later predictions. LSTMs are widely used in natural language processing, time series analysis, and other tasks involving sequential data.

2.3. Phase III: Evaluation metrics

In evaluating the tweet classification models, I incorporated six important metrics to ensure a comprehensive analysis. These metrics include Accuracy (ACC), F1-Score, Area Under the ROC Curve (AUC), Precision, Recall, and Specificity. By considering multiple criteria, I aimed to capture various aspects of the models' performance, such as overall correctness, balanced assessment, discriminatory ability, precision in positive predictions, recall for relevant instances, and specificity in recognizing negatives. This approach provides a well-rounded understanding of how effective the decision fusion models are in classifying tweets.

$$ACC = \left\{ \frac{(TP+TN)}{(TP+FP+TN+FN)} \right\} \in \{0,1\}$$

$$F1-\text{score} = \frac{2 \times P \times R}{P+R} \in \{0,1\}$$

$$AUC = \int_0^1 \left(\frac{TP}{TP+TN} \right) d \left(\frac{FP}{FP+FN} \right) = \int_0^1 \frac{TP}{P} d \frac{FP}{N}$$

$$\text{Precision} = \left\{ \frac{TP}{TP+FP} \right\}$$

$$\text{Recall} = \left\{ \frac{TP}{TP+FN} \right\}$$

2.3. Phase IV: Experimental Setup

I conducted all analyses on a MacBook Pro Air laptop equipped with an M1 processor and 64 GB RAM. Utilizing the Python programming language, I employed Scikit-learn, Pandas, and Seaborn libraries to perform the analyses.

3 Results and Discussions

3.1. Dataset Visualization

After following the methodology as discussed in the above section, all the preprocessing steps were done.

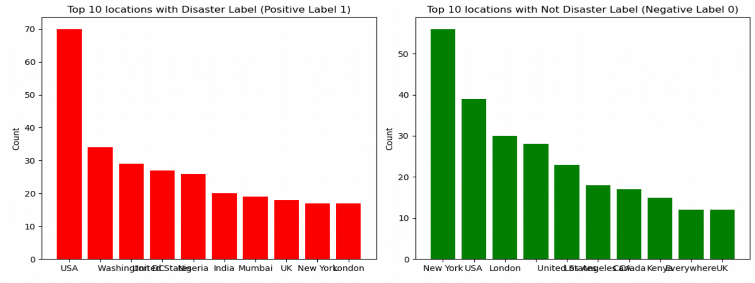


Figure 3: Top 10 locations for Disaster and Non-disaster

Figure 3 shows the top 10 locations in which disaster and non-disaster tweets were found.

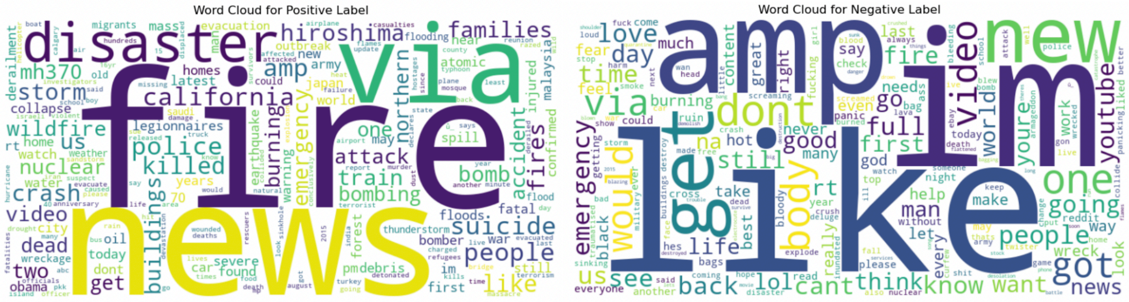


Figure 3: Positive & Negative Label

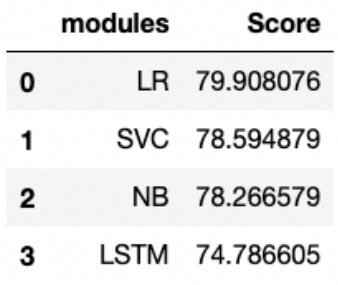
Figure 3 shows the Labels which were used for positive and negative label in disaster tweets. In this I used wordcloud library to find the labels in the dataset with max number of occurrences.As you can see in Positive Labels, “fire”, “news”,”via”,”disaster” are more widely used followed by “people”,”suicide”,”police”,”killed”,”families” and many more. Similarly, in Negative labels, “amp”,”im”,”like”,”new”,”get” are more widely used followed by “one”,”got”,”don’t” and many more.

3.2. Model Performance

In my analysis, Upon comparing all the models, a striking similarity in results emerges. Logistic regression, Support Vector Machine, Naive Bayes, and RNN with LSTM models exhibit accuracy levels ranging from 70% to 80%. Notably, Logistic regression outperforms the others, securing the top position with an accuracy of 79.90%. On the other end of the spectrum, RNN with LSTM records the lowest accuracy among the models, standing at 73%. This analysis indicates a consistent performance range across the various models, with Logistic regression demonstrating the highest predictive accuracy.

So clearly Logistic regression works well when predicting whether a tweet is referred to serious disaster events, or whether it is a metaphor tweet. In Logistic Regression, Recall stands at 88%, F1-Score is 83% where as Precision stands at 79%. In contrast, Recurrent neural network with LSTM produced least accuracy if 74%, with F1-Score as 78%, Precision as 77% and Recall as 77%.

Though the current analysis is confined to predefined hyperparameters, there exists the potential for improved accuracy by fine-tuning the hyperparameters of each model. Adjusting these parameters allows for a more nuanced exploration of the models' capabilities, opening up possibilities for enhanced performance and more accurate predictions.



	modules	Score
0	LR	79.908076
1	SVC	78.594879
2	NB	78.266579
3	LSTM	74.786605

Figure 4: Modules and its Accuracy

3.3 Comparison with related studies

I benchmarked the study against various research papers addressing similar models and problem definitions as shown in Figure 5. As you can see, many researchers find the accuracy to be mostly in the range of 70-80%.

S/N	Ref	Accuracy(%)
1	Humaid Alhammadi	79%
2	Berenice Jacqueline Sánchez Alvarado, Pedro E. Chavarrias-Solano.	79%
3	Shriya Goswamia , Debaditya Raychaudhurib	71%
4	This Study	79.9%

Figure 5: Related works and Accuracy

4. Conclusion

This research paper aimed to conduct text mining on Twitter datasets, specifically classifying tweets into two categories: disaster-related and not disaster-related. Based on the experimental findings, it can be confidently concluded that Twitter data can be effectively classified into these categories with a high accuracy rate.

To enhance the effectiveness of classifiers, it's advisable to explore alternative approaches to text mining and classification. The current method involves cleaning text data, establishing a corpus, and creating a Term-Document Matrix (TDM) to build models based on the TDM's relationship with the target variable. However, introducing context-aware methods, like incorporating variables such as word count or character count per tweet, could provide valuable insights and improve overall performance. Additionally, optimizing parameters for the chosen model, such as RNN with LSTM and considering other models like neural networks in Keras, may lead to increased accuracy. It's important to note that the time required for model building was identified as a challenge in the current approach.

Acknowledgments

I'm truly thankful to prof. Nti, who have been instrumental in bringing this study to life. His guidance and support have been invaluable throughout the entire process.

References

Berenice Jacqueline Sánchez Alvarado, Pedro E. Chavarrias-Solano. (2021). Detecting Disaster Tweets using a Natural Language Processing technique. https://www.researchgate.net/publication/356647529_Detecting_Disaster_Tweets_using_a_Natural_Language_Processing_technique

Humaid Alhammadi. (2022). Using Machine Learning in Disaster Tweets Classification. https://scholarworks.rit.edu/theses/11161/

Shriya Goswamia , Debaditya Raychaudhurib. (2020). Identification of Disaster-related tweets using Natural Language Processing. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3610676

Addison Howard, devrishi, Phil Culliton, Yufeng Guo. (2019). Natural Language Processing with Disaster Tweets. Kaggle. https://kaggle.com/competitions/nlp-getting-started