Machine Learning & Deep Learning: Classifying Disaster Tweets Methods

Dr Manisha Mali[1], Rohan Nadekar[2], Saif Kumbay[3], Najim Tadvi[4], Jayesh Walke[5], Arnav Mone[6]

[1,2,3,4,5,6]Department of Computer Engineering, VIIT, Pune, India

Abstract
With the increasing reliance on social media sites, and more importantly on Twitter, during disaster events, there is a strong need for an immediate capability to effectively analyze large quantities of near-real-time data into meaningful information in support of more efficient emergency responses. This paper reviews new trends in disaster tweet classification using machine learning and deep learning. We have picked five major studies with approaches ranging from more basic approaches to more complex approaches with Logistic Regression and Naive Bayes to using CNN and BERT deep architectures. Findings of this study include apparent flips in the use of BERT embeddings for improving accuracy and relevance of disaster-related tweet classification, which essentially provides emergency responders with timely and critical information when crises arise.. Review work calls for the input of advanced deep learning techniques to answer the complexities involved with the real-time data streaming that characterizes a disaster scenario.

Keywords: Disaster response, tweet classification, machine learning, deep learning, BERT embeddings, social media analytics.

Introduction

In the last few years, because of the growing popularity of social media networks, especially Twitter, data analysis related to social media, especially tweets, has become one of the most significant components of disaster management and emergency response. A great deal of information was recovered across social media networks in times of disasters in terms of public mood, resource needs, and issues of immediate nature. However, because of the magnitude of data, it was challenging in many cases to provide the right information in the right amount of time to emergency responders. Consequently, using machine learning and deep learning techniques in the classification of disaster-related tweets has garnered immense attention. This paper aims to assess the performance of a few computational models for the task of disaster tweet classification mainly focusing on recent advances in methodologies for deep learning. Traditional machine learning algorithms: Logistic Regression, Naive Bayes, and SVM, act as a foundation of this field. Most notably, works by Islam et al. [1] strengthened the notion that Twitter plays a very important role as an information source within disaster events but subdue the advantage of traditional approaches in the capture of semantic richness of tweets.
In order to overcome this limitation, the recent studies emphasize fusion with state-of-art techniques like CNN and BERT embeddings. For instance, Dharrao et al.

In [2], it was demonstrated that a CNN model with BERT embeddings: Improved classification Accuracies and how deep learning may access Contextual relations that exist in disaster-related tweets.Manimegalai et al. [3] extended that by using a combination of CNN and RNN along with BERT. This, in fact, demonstrated how with such a model, real-time data could be processed while distinguishing between what is relevant and irrelevant information. Optimization methods have also been explored for BERT-based architectures, including Le et al. [4], who claim that tuning of hyperparameters could further enhance the performance of the model. The experimental results they report demonstrate the contextual embeddings supplied by BERT are very much more superior than conventional word embedding techniques like TF IDF, in the event of disaster tweet classification as well. Additionally, Jiang et al. [5] have added some strength to the argument made for using BERT in conjunction with LSTM networks with examples of its ability to understand both forward and backward contexts. In a word, disaster tweet analysis evolved from the traditional machine learning approaches to more complex deep learning techniques, changing the field in the most radical way. The studied literatures have thus explored and brought forward an integral idea that, within the development of BERT embodiments, increases the accuracy and pertinence of disaster-tweet classification. Thus, the focus of the presented paper supports the ongoing debate on this by looking at the current methodologies used within the implementation of necessary deep learning techniques for enhanced emergency response.

LITERATURE REVIEW

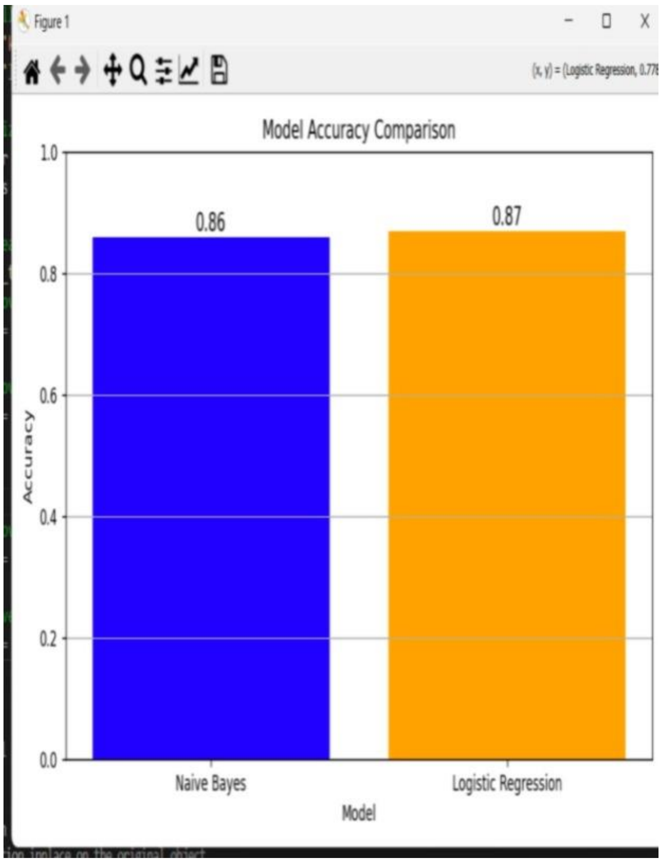
This area of disaster tweet analysis using machine learning \ and deep learning has grown rapidly and concentrated more \ on the effective extraction of meaningful information from \ the massive volume of social media data in events of \ emergencies. This paper reviews five key papers that have \ different approaches for improving the accuracy and \ relevance of disaster tweet classification.

Islam et al. [1] laid a sound foundation for the usage of \ machine learning to classify disaster tweets \ as their method used the Logistic Regression, \ Naive Bayes, and SVM models on the tweet dataset \ collected during hurricane and earthquake-related events. Their \study revealed the potential importance of Twitter as an emerging source \ of real-time information during disaster events, and shed insight into the effectiveness with which conventional machine learning models might be used for classifying the tweets as either disasters or non-disasters. The logistic regression was better and achieved an accuracy rate of 80.5% more than other models. The study noted that there is a need to have effective filtering of tweets to support the identification of information that would be Fastly available to emergency responders. However, research acknowledged a lack of model performance; deep learning techniques also have potential further improvements in the semantical capture from the tweet data.

From the above findings, Dharrao et al. [2] discussed developing a CNN with BERT embeddings and RMSprop optimization deep-learning-based approach for improving classification tasks on tweets, considering that traditional machine learning approaches are incompetent in robustly capturing nuances and contextual relationships inherent to disaster-related tweets. BERT embeddings were first introduced into this work through their incorporation into a CNN framework to capture the contextual meaning of words based on their both left and right contexts. Their model achieved an accuracy of about 83% with an F1-score of 0.80, significantly better than that of traditional models. This study proved that the deep learning models, especially BERT embeddings, can give more accurate and reliable classification in tweets to assist the disaster response teams get easy access to relevant information.

Manimegalai et al. [3] took it further exploring the applications of BERT embeddings in classifying disaster tweet by combining with both CNN and Recurrent Neural Networks (RNN) architect

From the description, I see it looks like some kind of data table with columns id, target, location, and text. The cold data description of how it looks is as follows:
id: A unique number for each file.
target: 0 or 1 - a binary value, could be used to stand in for a classification or sentiment for an event (e.g., something that is serious or of an emergency, etc., in relation to not being casual or urgent).
location: shows the location related to every entry, e.g., USA, IND, China, etc.
Each of the entry has a descriptive text sometimes of accidents, natural disasters, and conversations.
Important few points of the data:
Serious Events:
Line 2: car crash-USA
Line 3: earthquake warning
Line 4,5: forest fire in India, fires in Spokane (USA)
Line 6: Typhoon Soudelor in China and Taiwan
Line 7: people experiencing the shaking of an earthquake in India
Casual/Conversational Entries:
Line 10 says, "Somebody asked after Arsenal football.
Lines 11-14 are conversational like "Hey! How are you?", "What a nice hat?", and "Fuck off!"

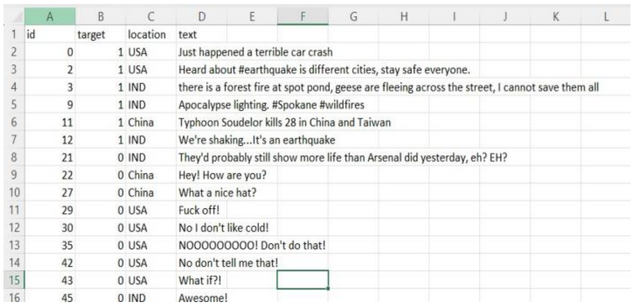


(FIG:1)-Comparison of Model

ures. Their study proved the flexibility of BERT in interpreting the depth of semantics of tweets so that relevant and irrelevant information regarding disasters may be distinguished more efficiently than the traditional models. Their experiments showed that the application of BERT embeddings boosts the classification of tweets immensely, making the model efficient in processing real-time data on Twitter. The authors pointed out that the ability of their model to account both contextual and temporal aspects of the tweets made their model a powerful one for disaster response efforts. This composite technique in deep learning allowed a far better predictive capability of disaster tweets; thus, emergency response teams can make better decisions.

Le et al. [4] furthered the analysis by studying fine-tuning strategies for BERT-based models for the optimization of disaster tweet classification. They are using a Kaggle dataset of more than 10,000 labeled tweets and experimenting with several optimization algorithms, such as RMSprop and Adam, to maximize the performance of the model. Their results indicated that BERT embeddings outperform the conventional word representation methods such as TF-IDF and Count Vector in comparison with the traditional word representation techniques for the classification of disasters and reach an F1-score greater than 80%.The study also points out that for higher model performance, hyperparameter tuning is needed in the training process. Le et al. concluded that BERT's contextual embeddings offer a tremendous advantage in dealing with the complex nature of disaster-related tweets, usually very short and sensitive to context.

In the final analysis, Jiang et al. [5] explored BERT-based embedding together with Bi-LSTM networks in order to



id	target	location	text
0	1	USA	Just happened a terrible car crash
2	1	USA	Heard about #earthquake is different cities, stay safe everyone.
3	1	IND	there is a forest fire at spot pond, geese are fleeing across the street, I cannot save them all
9	1	IND	Apocalypse lighting. #Spokane #wildfires
11	1	China	Typhoon Soudelor kills 28 in China and Taiwan
12	1	IND	We're shaking...It's an earthquake
21	0	IND	They'd probably still show more life than Arsenal did yesterday, eh? EH?
22	0	China	Hey! How are you?
27	0	China	What a nice hat?
29	0	USA	Fuck off!
30	0	USA	No I don't like cold!
35	0	USA	NOOOOOOOOO! Don't do that!
42	0	USA	No don't tell me that!
43	0	USA	What if?!
45	0	IND	Awesome!

(FIG;2)-Datasets

classify disaster-related tweets. They compared the BERT-based embeddings with old word-based embeddings like GloVe and Word2Vec and easily found that BERT simply outperformed these older techniques by a huge margin. The Bi-LSTM network captures the forward and backward contexts of words, which worked synergistically with BERT to provide superior results over disaster tweet classification. Their model offered better accuracy and F1 scores than the previous models and underscores further the use of deep learning models using contextual embeddings in tasks related to disasters. The five studies above further go on to exemplify the necessity of using advanced deep learning techniques, especially with BERT, in order to adequately process and classify real-time data streaming in from Twitter in the wake of a disaster.

Overall, the five works above constitute a continuous movement into new usage of newer deeper learning techniques that supplant old techniques of machine learning when used for classifying disaster tweets. BERT's utilization in the studies revealed how context-aware models constitute part and parcel in enhancing accuracy as well as relevance while classifying the tweets. Such studies have significantly amplified performance in disaster-related tweet classification, where their findings are highly useful tools for emergency response teams to rely on accurate and timely information in a time of crisis.

1. A. Text Data Normalization Standardize the text data by applying data cleaning methods like removing URLs, mentions, hashtags, punctuation, and stopwords, which ultimately enhances the model in the end

2. B. TF-IDF Transform the Term Frequency-Inverse Document Frequency 1234 transformation to the cleaned-up text data, encode the data as numeric features where the significance of unique words would be considered important, and frequent words would reduce.
3. D. Compare Model Performance: In order to compare classification accuracy as well as other performance metrics of two different models, train and evaluate two different models: Multinomial Naive Bayes and Logistic Regression on the same dataset.
4. E. Evaluate Evaluation Metrics: Such critical models must be evaluated with regard to some appropriate metrics; for example, accuracy, precision, recall, and F1-score, representing the performance of the model in providing tweets regarding a disaster.
5. F. Performance of Models to be Visualized: All the models' performance metrics need to be represented in graphs and confusion matrices that will help further understand the extent of performance of each model and areas where one needs to work for improvement.
6. G. To Provide Insights for Future Research: This is derived from the outcome of what can be drawn from the results obtained from the performance of the models. This can provide areas of further research and future work, therefore contributing to the overarching theme of disaster response and social media analytics..

METHOLOGY

The approach to the project consists of a number of key stages, starting with data collection and preprocessing. I depended on two datasets: `train.csv` and `test.csv`, covering information regarding tweet text, keywords, location, etc. with a target label indicating whether the tweet is related to disaster or not. Preprocessing The textual data cleaned URL, special character, punctuation, and stopwords through lemmatization. There are also missing values in 'keyword' and 'location'. So, all missing values in these columns were filled by an empty string; rows that had missing 'text' or 'target' values were deleted.It then followed the process of feature extraction using the TF-IDF (Term
Frequency-Inverse Document Frequency) vectorizer that transforms text data into numerical features. The TF-IDF
matrix catches the words of importance for the tweets, giving emphasis on the more discriminative ones. Two machine learning models were then trained for this experiment: Naive Bayes and Logistic Regression. Data is divided into two sets : training and validation set. Page 12 of 19 - Integrity Submission Submission ID trn:oid:::3618:69197613 Page 13 of 19 - Integrity Submission the performance of the model. To compare, the accuracy, precision, recall, and F1-score score for each model was used.
. Finally, predictions are made on the `test.csv` dataset and a new CSV file with the predicted `target` values is created so that it can be further compared. This entire process ensures that there is a systemic comparison of both the algorithms in identification of disaster-related tweets.

A. 1. Data collection and Preprocessing

Dataset: We used two sets of data for this research work: train.csv and test.csv. These datasets were accessed for the purposes of the different intents in the learning algorithm pipeline - training and testing. These data sets included tweets which were relevant to the keywords from the metadata, including location, and whether it had relevance to a disaster or not.

1)Train.csv:

- There are five columns for the training dataset: id, keyword, location, text, and target.
- 'id': This is an ID for each tweet that helps in differentiating the records.

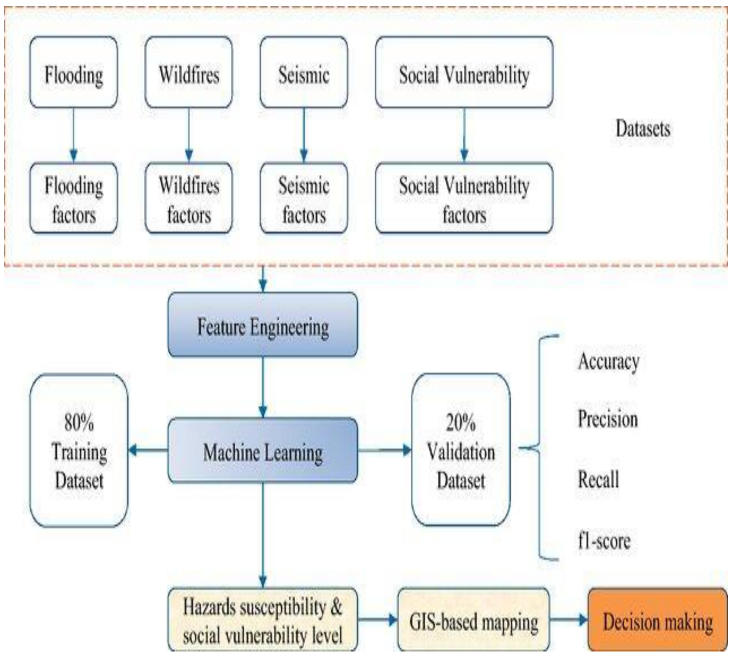
- keyword: This column includes disaster-related words or phrases that occurred in the tweet, which gives some contextual information about why a given tweet would be classified as disaster-related. Often, these keywords bring attention to the nature of the tweet (e.g., "earthquake," "flood").
- location: This field provides the geographic location of the tweet. Such information can offer additional context, especially if the tweet originates from an area affected by a disaster.
- text: This column holds the raw text of the tweet, which serves as the primary input for most Natural Language Processing (NLP) models. The text often contains information relevant to disasters, either directly or indirectly.
- target: This column will be the label for each tweet, indicating if the given tweet is related to a disaster or not. It allows the models to be trained in a supervised learning fashion and points out patterns between disaster-related and non-disaster-related tweets.

The training dataset is important because it provides labeled data through which machine learning models can learn the patterns and relationships underlying the tweet text to predict disaster-related tweets.

Test.csv:

The test dataset has a similar structure to the training dataset but lacks the target column. It includes four columns: id, keyword, location, and text.

This dataset is used for making predictions after models have been trained. Since it does not contain thetarget labels, t



(FIG:3)-Flow Diagram

The goal is to predict this missing label for all tweets—indicating whether they are disaster-related or not. The models used are Naive Bayes and Logistic Regression, trained and evaluated on the training dataset. Upon completion of the training, a prediction is done on the test dataset. The results will be used to generate a new CSV file containing only the id column from the test dataset and the predicted target column. Both datasets contain missing values, inconsistent entries in the keyword and location columns, and noisy text in the text column. Therefore, preprocessing of noisy texts is necessary before utilizing the data for model training.

Data Cleaning

To clean the text data for this project and effectively interpret and classify the tweets, the following activities have been undertaken. Many texts of raw tweets may often contain unwanted elements that will introduce noise and therefore influence the performance of a machine learning model. Multiple steps of preprocessing were used on the data to fashion the data into a suitable analysis format and to classify them correctly. The following gives descriptions of each step and the relevance of these steps.

This application will use a regular expression to remove URLs, mentions, and hashtags from the given text.

Many of the tweets contain URLs, mentions (@username), and hashtags (#disaster), which often do not contribute meaningful information. These elements are platform-specific and act as noise, irrelevant to the task of disaster tweet classification.

- URLs: URLs link to external content and do not provide semantic meaning within the tweet. Removing them helps prevent the model from treating these characters as important features.
- Mentions: User mentions (e.g., @someone) are generally irrelevant for classification and removing them helps avoid overfitting to personalized, context-specific information.
- Hashtags: While hashtags can contain disaster-related keywords, the keyword column in the dataset already captures these phrases. Removing hashtag symbols (e.g., #) retaining the keyword enhances readability and analysis.
- Regex is used to find and remove unwanted elements in the text. Regex is a powerful tool for pattern matching that can be very effective in preprocessing texts.
- All Lower Case to Normalize Text

  Converting all text to lowercase ensures consistency across the dataset. For example, "Flood" and "flood" should be considered the same word, but without normalization, they would be treated as separate tokens. Lowercasing reduces variability in the data and improves model performance, as case differences do not influence the interpretation of the text.
- Removing Punctuation and Stopwords

  Punctuation: Punctuation characters (e.g., periods, commas, exclamation points) do not carry semantic value in the text. Stripping these characters reduces noise and helps the model focus on words rather than irrelevant symbols.
- •\\tStopwords: Words like "the," "is," and "and" appear the most in text but are of little use in classification. Removing stopwords minimizes dimensionality and increases the signal-to-noise ratio. NLTK was used to remove stopwords.By eliminating stopwords and punctuation, the model focuses on meaningful words, ultimately improving its ability to classify disaster-related tweets.
- Tokenizing Words and Applying Lemmatization

  Tokenization: Tokenization refers to breaking a sentence into individual words or tokens. For instance, the sentence "Floods in India are devastating" would be tokenized as ["Floods," "in," "India," "are," "devastating"]. This step simplifies the text for easier analysis by the model.

•\\tLemmatization: Lemmatization is the process of bringing words to their base or root forms. For example, "running" gets lemmatized to "run," and "better" to "good." Lemmatization considers different forms of a word to be the same feature that reduces text complexity.

In this exercise, I opted for WordNetLemmatizer found in the NLTK library. Lemmatization is a method distinct from stemming because it provides more meaning of the original word; it converts words into meaningful root forms.

These cleaning steps are critical to enhance the performance of the model to correctly classify the tweets. Noise elements such as URLs, mentions, and punctuation reduce the accuracy of the model. Removing stopwords helps the model focus on meaningful features. Tokenization and lemmatization reduce the variability in the text, and this helps the model find the underlying linguistic patterns.
All of these pre-processing techniques standardize, clean, and simplify text data so it can become more meaningful for the application of Naive Bayes and Logistic Regression classifiers. The cleaning up of this text data tends to generalize patterns across the dataset properly with higher precision and accuracy. The main tools employed in this pre-processing pipeline were the NLTK library and regular expressions.

B. Feature Extraction: TF-IDF Vectorization

- In this assignment, the text data had already been converted into numerical features using the Term Frequency-Inverse Document Frequency method. Here, any machine learning model requires numerical input. Any raw text needs to get converted into a numerical representation because TF-IDF is used as an effective technique here.
- TF-IDF: Concept and Formula
- TF-IDF represents Term Frequency-Inverse Document Frequency. It is the numerical value that should statistically measure a term in document significance within a given general collection, often called corpus. The actual mathematical expression given for computation of value TF-IDF for the word t on document d follows the one below;
- TF-IDF(t,d)=TF(t,d)×IDF(t)
- where,
- TF: Term Frequency; A term is just how frequent word t comes out appearing in the d document
- TF(t,d)= Appearance of t in d/ Total terms in d
- IDF: Inverse Document Frequency is the measurement for how unique or how less common a word is on the whole corpus. This way, rare terms will carry higher weights and lower ones frequent ones.
- IDF(t)= log⁡〖((Total number of documents)/(Number of documents containing t))〗
- In simple words, it gives the importance of the term in a single tweet and IDF reduces terms, which are common across all tweets giving higher importance to terms appearing in fewer number of tweets. Thus the final TF-IDF score for each word is combination of both, thereby catching the relevance of a term with the context of each word with penalty on frequently occurring terms that add little value.
- Why TF-IDF?
- There are several reasons why I have chosen to use TF-IDF over Bag of Words.
- Down-plays Frequently Occurring Words: Unlike the technique Bag of Words that treats words as if they were identical, it reduces the impact of words such as "disaster" or "help," occurring frequently across all the tweets but which don't help in deciding that whether a related tweet or a not, is disaster-related.Highlights Rare, Precise Words: Terms appearing very seldom but being very closely related to some tweets (for instance, "hurricane," "earthquake") are weighted more heavily by the IDF component. These are often the most informative words for determining content related to disasters.
- Normalization: The TF-IDF normalization process automatically normalizes the term frequencies, so longer tweets do not get an undue advantage by containing more words than shorter ones.

Selecting max_features = 5,000
We constrain the maximum number of features to be 5,000. The choice is motivated by the necessity to seek a trade-off between computation efficiency and the ability to catch the most relevant words, keeping

in mind the optimal trade-off: limiting the feature space helps to reduce overfitting:

- By capping the number of features, it prevents learning rare and extraneous terms, occurring only once or twice in the entire corpus.
- A smaller feature space will improve the speed of training and the model from overloading on too many input variables, yet still capture the most relevant terms as determined by the TF-IDF scores.

Selecting 5,000 features will ensure that we only bring in the top 5,000 most significant terms in the text, thereby capturing a wide informative vocabulary without adding unnecessary noise in the data.

Importance for Classification

TF-IDF therefore converts the cleaned up text of the tweets to a readable format for machine learning algorithms. This would be achieved by casting each one of the tweets as sparse matrices. Here, a row represents a tweet while the column would represent each of the top 5,000 terms from the corpus. Then, in the values of the matrix, we put the score of the word TF-IDF of each word found in that specific tweet.

It is in this vectorized representation that Naive Bayes and Logistic Regression models spring up as possible since it is showing that the text data, with numerical modeling, signifies features that would distinguish or make a difference between disasters related to tweets and that which is not so.

By using TF-IDF, we are ensuring that the models focus on the most meaningful terms in the files-this in turn promotes better predictions and classification accuracy. Overall, striking a balance between term frequency and rarity through TF-IDF happens to present a particularly natural fit for the context of text classification in tasks of this nature, whereby some words are more pivotal than others for distinguishing between the target classes.

C. Model Selection

We focus, in this sections, on two of the dominant models applied in our analytics work on disaster-related tweets: Multinomial Naive Bayes and Logistic Regression. Each model has benefits that suit the nature of the dataset as well as the classification task involved.

Multinomial Naive Bayes

The Multinomial Naive Bayes algorithm is a type of probabilistic classifier especially designed for text classification. The most important point about Naive Bayes is the application of Bayes' theorem, which states that:

nn$P(Y|X)= (P(X|Y)\cdot P(Y))/(P(X))$

Where:

$P(Y|X)$ is the posterior probability of class Y given the features X

$P(X|Y)$ is the likelihood of the features X given class Y.

$P(Y)$ is the prior probability of class Y.

Let the random variable X represent all feature set and $P(X)$ represents prior probability of features X.

In the text classification features X are words in a document and class Y be 0 if the tweet is not a disaster; else, it is equal to 1 (if the tweet is a disaster).

- Since word frequencies in a text are discrete, Multinomial Naive Bayes classifies features as a multinomial distribution. The class Y, given a document d, can be written as:
- n$P(Y|d) \propto P(Y)$i=∏_(i=1)^n▒P(wi|Y)
- Where:
- P(wi|Y) is the probability of the word wiw_given the class Y.
- n is the number of words in the document.
- Why Naive Bayes is well suited for text classification:

- Dealing with discrete features: The multinomial Naive Bayes deals easily and effectively with discrete features. This is because text data are usually given as counts, where each feature is a count or frequency.
- Simplicity and Speed: The model is computationally efficient in the sense that it involves very few parameters that must be estimated, thus leading to rapid training and prediction even on large datasets.
- \\tIrrelevance to the feature: Naive Bayes is robust against the presence of irrelevant features. Since each feature is taken as independent, the assumption of independence in the model makes it light and decreases noisy data impact.

The Multinomial Naive Bayes model acquired will be trained using TF-IDF features obtained from cleaned tweet text. This model could possibly be capable enough to predict whether a particular tweet belongs to the disaster or not.

Logistic Regression

Logistic Regression is a statistical model. It predicts the likelihood of a binary outcome in relation to one or more predictor variables. In this case, it predicts the chances of a tweet being related to a disaster (1) and not related to a disaster (0). The logistic function, also known as the sigmoid function, maps the predicted values to probabilities:

P(Y=1|X)= 1/(1+ e^(-z))

Where:

z=β0+β1x1+β2x2+…+βnxn

xix_ixi feature (TF-IDF weights) and βi:β_iβi- coefficients to be learned after training the model.

Optimal coefficients β : such that the likelihood of being able to observe the assigned data is maximized (usually through maximum likelihood estimator methods).

Justification to include in the Study:Interpretability: Logistic Regression is highly interpretable. This simply means that it is easier for the researchers to understand the effect that each of the individual features has on the outcome. That clarity is good in understanding how the specific terms relate to disaster-related tweets.Benchmarking: Using Logistic Regression in combination with the Multinomial Naive Bayes provides the opportunity to have a baseline that performance could be compared against. Logistic Regression would serve as the easier baseline against which performance could be compared to more complex Naive Bayes.

A Simple Model's Effectiveness. Although very simple, logistic regression can easily give surprisingly good results with proper definitions of features, such as TF-IDF scores. Including it in our pipeline allows us to verify how lightweight linear models stand on text classification tasks and what their strengths and weaknesses are.

In summary, both Multinomial Naive Bayes and Logistic Regression were strategically chosen due to the strength of each model on text data for this analysis. The comparison between these two models not only highlights how both models performed well in classifying disaster-related tweets but also yields some useful information about the applicability of such algorithms in natural language processing tasks.

Training and Testing Process: Splitting the Dataset

D. The data was split into two large sets, namely a training set and a validation set, to accurately test our classification models. This was done with an 80-20 split. In other words, 80% of the training dataset was used for training the models, while the remaining 20% was used to evaluate the performance of the models. This stratified splitting ensures that both subsets retain a close distribution of disaster-related and non-disaster-related tweets, thus allowing for more dependable performance evaluation.

D. Why Validation Set?

E. A validation set is required to get an estimation of the ability of generalization of these models. Therefore, there exists a scope to note the performance of models about the unseen data and so that there is a much less probability that models only memorize the training dataset. This is termed generally as overfitting where such a model learns very high accuracy over the training but does not generalize for examples of a new, unseen phenomenon. We can thus obtain an unbiased estimate of how the models will eventually perform in real life based on their performances on the validation set that guides further refinements and adjustments.

F. Model Training

Now, the Multinomial Naive Bayes and Logistic Regression were trained on the training set derived on the basis of the raw dataset. For this problem, such training will fit the models along with transformed TF-IDF features from the tweets in a manner that those two models understand which patterns distinguish the disaster-related tweets from those which were not related to any disaster.

1. Training of Multinomial Naive Bayes Model: A Multinomial Naive Bayes classifier has been trained with and learned from the features of TF-IDF on the training set. This training period calculates the prior probabilities for classes along with conditional probabilities for all the features, that is every word for the specific classes. The model so fitted in data learns how to classify what kind of word distribution ideally would be considered what kind of tweet.

2Training the Logistic Regression Model: The Logistic Regression model is created using the same training set. In this example, the model has utilized the sigmoid function to predict probabilities of disaster-related tweets. Optimization techniques such as gradient descent are used in order to learn coefficients for each feature maximizing the likelihood of the training data given the predicted probabilities.

Hyperparameter Settings: Of the Multinomial Naive Bayes model, hyperparameter tuning was minimal because the algorithm relies on very simple parameter settings such as smoothing parameters, for example, Laplace smoothing. In this study, Laplace smoothing was applied at the default value of 1 to prevent zero probabilities for unseen words on the validation set. In the Logistic Regression model, we have tried different strengths of regularization and types of regularization. We used grid search for the right values of these hyperparameters to make the model robust and avoid overfitting.

This splitting of the dataset into these two parts will be careful and systematic in both the training phases of the models, in which it is a good guarantee for a complete performance analysis of both models so that they can predict validly and classify disaster-related tweets. Through such techniques, meaningful insights are drawn into the efficacy of each model in solving the research problem at hand.

Evaluation Metrics

- In the classification models for tweets about disasters, some of the most key metrics used were accuracy, precision, recall, and the F1-Score. These not only give a better view of the kind of performance each model performs but may also be used to establish whether the classification would distinguish between disaster-related and non-disaster-related tweets.
- Accuracy: The simple measure of overall performance of a classification model is accuracy, which is defined as the quality of its predictions in comparison to total number of predictions that model has provided. Mathematically we can define this as follows:
- Accuracy= (True Positive+True Negatives)/(Total Predictions)
- Here,

- True Positives (TP): Number of the disaster-related tweets identified correctly by the model
- True Negatives (TN): Number of the non-disaster-related tweets identified correctly by the model.
- Total Predictions: Sum of all the predicted instances including true positives, true negatives, false positives, and false negatives.
- Accuracy gives a rough estimate of how good your model is, but that still does not tell you much about whether your model is going to work in the imbalanced dataset scenario, for instance, where there are many more non-disaster related tweets than disaster-related ones.

Recall: Recall or sensitivity measures the number of true positives the model identified. Recall is defined as the ratio of actual positive results to the total number of actual positives:

Recall= (True Positives)/(True Positives+False Negatives)

In disaster detection, high recall is significant as it ensures most disaster-related tweets are found. Failure to detect those tweets may lead to serious results, for instance, delayed response to emergencies.

Actually, the very practical nature of disaster detection necessitates high recall: most of the tweets related to disasters must be identified. Failure to detect some of these can lead to very dire consequences, including delayed emergency responses.

$F1=2 \times (Precision \times Recall)/(Precision+Recall)$

Therefore, F1-score is a good measure as it does really very well over both the precision and recall metrics of a model for some application of potential disaster detection.

Disaster detection involves a game of high stakes since inappropriate response at emergency times due to false positives regarding disaster-related tweets can bring terrible after-effects like spread of misinformation. As a result, accuracy measures alone may hide many very important insights into the model's performance. We now see how to look at precision, recall, and the F1-Score-by looking at these three aspects, it is quite easy to assess how good a model is in the process of ensuring that we actually do the best with respect to maximizing relevant disaster-related tweets and minimizing false alarms. Such a multi-faceted approach in terms of evaluation contributes to a much stronger understanding of the capabilities and limitations of the models within this domain of disaster detection.

Conclusion: The methodology that we applied to this project gave us a nice comprehensive framework for the analysis of the tweets and determination of its potential relevance to a disaster event. By proper data gathering and preprocessing, we cleaned the text data for further analysis. Feature extraction via TF-IDF vectorization converts textual information into meaningful numerical representations that allow effective application of machine learning algorithms. Having both the Multinomial Naive Bayes and Logistic Regression models selected, it forms a strong base for comparison to make careful judgments about the actual performance using accuracy, precision, recall, and F1 score metrics. This multifaceted approach is quite reliable in terms of the reliability of our predictions but brings into fore the needs for careful model evaluation within the sensitive domain of disaster detection.

Result

In this paper, two classification models applied for the classification of disaster-related tweets are Multinomial Naive Bayes and Logistic Regression, based on a labeled tweets dataset. Before starting to create the classification models themselves, the dataset has to be cleaned of noise through the removal of URL and special characters and furthering into token lemmatization and stopword removal. For proper robust estimation of the performance of models, There is an 80-20 training vs. validation split on this dataset.

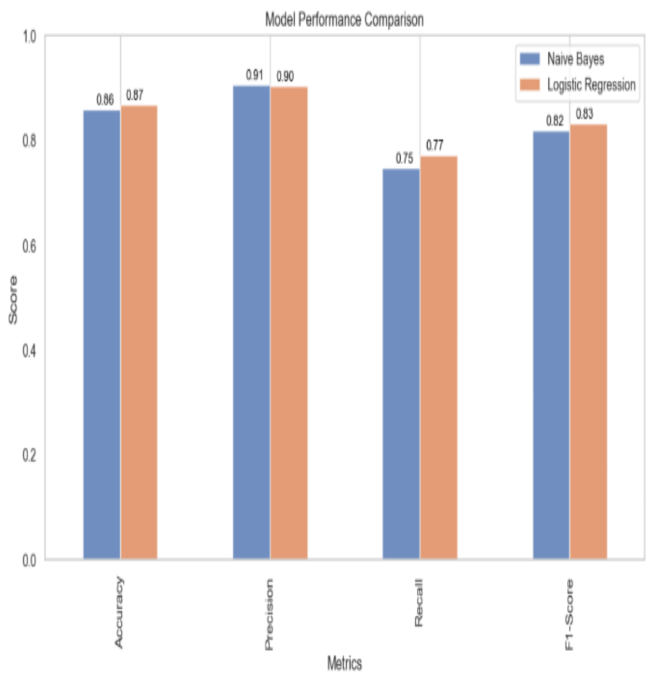
Multinomial Naive Bayes and the Logistic Regression model has been trained on TF-IDF features of the clean text.
Accuracy, precision, recall, and F1-score are the measures to assess their performance. Classification report, for both models, generated during this process is very effective in drawing individual insights from them.

1. Multinomial Naive Bayes Model Performance:

- Accuracy: 85.9%
- Precision: 90.65%
- Recall: 74.73%
- F1-Score: 81.92%

2. Logistic Regression Model Performance:

- Accuracy: 86.80%
- Precision: 90.43%
- Recall: 77.19%
- F1-Score: 83.29%



Fig(4)Performance Comparison

The accuracy scores that you would have obtained for the Naive Bayes and Logistic Regression models indicate that both models did fairly well in terms of predicting disaster-related tweets, although Logistic Regression outperformed Naive Bayes with a very slight margin. Let us break down some reasons behind these accuracy results:

- **Reasons for Accuracy Scores**

Naive Bayes: It actually assumes features are independent and something that may not happen in real data. Though it might be good for text classification, it would miss relationships of words that could be important for the context, flood, earthquake.
Logistic Regression: This will capture linear relationship; therefore, it can be imagined to capture a relation that can be combined as different terms in order to predict disaster relevance. That might then increase the chances of accuracy because the words in the tweets could have interdependencies.

- Feature Representation: TF-IDF Vectorization: Same feature set produced using the TF-IDF method is utilized in the two models, but since Logistic Regression fits a line or hyperplane in a space of dimensions, it is likely to utilize the representation better for its improvement. It may also optimize terms that are potent in predicting disaster tweets.
- Overfitting vs. Underfitting: Since Naive Bayes uses strong independence assumptions, it might lead to underfitting the data and thereby may provide lower accuracy compared to Logistic Regression, which may fit the data properly without major overfitting.
- Distribution of Data: Performance may also depend on the type of dataset. Perhaps the disaster-related tweets are distributed in a manner that certain characteristics, or keywords, are better suited for Logistic Regression.
- Hyperparameters and Settings: If you have ran some hyperparameters on top for the Logistic Regression model (such as regularization), it may provide better performance. On the other hand, if you have not tuned any parameters for Naive Bayes, the default configuration will not necessarily be the best for your dataset.

Therefore, the results indicate that both of these models are able to work for this task but with a slightly higher edge to Logistic Regression because of the marginally higher accuracy, probably since the former can detect more complex relationships in the data. If you would like to possibly get even better results for the performance of the model, then you can try some feature engineering, hyperparameter search, or even ensemble methods such as Random Forest or Gradient Boosting.

Conclusion

When looking at tweets about disasters, Naive Bayes and Logistic Regression are good contenders that fit the scope of a project. Naive Bayes is especially favored because it is very simple, yet very effective in its application in areas where an immediate outcome is required but with a small amount of computing resources. It performs well in applications such as text categorization especially with the use of methods, such as bag of words or TF IDF, and can handle data sizes. The assumption that features are independent may degrade its performance, where in datasets words and features are interlinked and their results affect each other. It sometimes produces less accurate results. Contrary to this is Logistic Regression. Although it consumes resources, it takes a subtle strategy considering the relationships between the features without any assumption of independence. Logistic Regression outshines in handling correlated features. Often outperforms Naive Bayes, especially in terms of higher accuracy levels as well as precision and recall rates. Here, Logistic Regression stands as a choice when precision and its ability to capture data patterns become the pivotal considerations for making the choice. While Logistic Regression might be slower and more resource-hungry compared to other models like Naive Bayes or Decision Trees; the beauty of logistic regression is its ability to classify disaster-related tweets with the least possible false positives and negatives, which makes it an obvious choice for complex or critical tasks. In the end, it depends on which balance to be reached between the speed and simplicity on one side and accuracy and complexity on the other side.

References

1. Islam, I. Hossain Pranto, M. Islam, M. T. Mahmud, R. Rudra, and M. M. Foysal, "Optimized Convolutional Neural Networks Enhanced with BERT Embeddings for Quick Disaster Tweets Classification.," INCOFT 2024, February 2024, pp. 1-8. DOI: 10.1109/INCOFT60753.2023.10425631
2. D. Dharrao, A. MR, R. Mital, A. Vengali, M. Pangavhane, S. Rajput, and A. M. Bongale, "An Efficient Method for Disaster Tweets Classification Using Gradient-Based Optimized Convolutional Neural Networks with BERT Embeddings," MethodsX, vol. 13, pp. 102843, 2024. DOI: 10.1016/j.mex.2024.102843.
3. R. Manimegalai, G. V. S. R. Jegan, S. Arawind, and B. Gomathi, "Dtweet: Disaster Tweet Analysis Using Deep Learning Techniques," Springer, 2023. DOI: 10.1007/978-981-99-1881-2_14.
4. A. D. Le, "Disaster Tweets Classification using BERT-Based Language Model," arXiv, 2022. DOI: arXiv:2202.00795.
5. C. Jiang, Y. Chen, and S. Chen, "Efficacy of BERT embeddings on predicting disaster from Twitter data," arXiv, 2021. DOI: arXiv:2108.10698.
6. Paul, Classification of crisis-related data on Twitter using a deep learning-based framework, Multimed. Tools. Appl., № c. 8921.
7. \tKarimiziarani, Social response and Disaster management: insights from twitter data Assimilation on Hurricane Ian, Int. J. Disaster Risk Reduct., № 95.
8. tLi, Disaster response aided by tweet classification with a domain adaptation approach, J. Contingencies Crisis Manage., № 26, c. 16.
9. \tMadichetty, Detection of situational information from Twitter during disaster using deep learning models, Sādhanā, № 45, c. 270.
10. Krishna A, Deep parallel hybrid fusion model for disaster tweet classification on Twitter data, Decis. Anal. J.
11. Prasad Transport disaster tweets identification and categorization using improved Bidirectional Encoder Representations from Transformers, Int. J. Infor. Manage. Data Insights, №3 12 Gite Ant-colony optimization for extracting the textual features in the framework of hate speech classification, Big. Data Cogn. Comput., № 7, c.
12. \\tTalaat, Sentiment analysis classification system using hybrid BERT models, J. Big. Data, № 10, c. 110
13. \\tZaman, A comparative analysis of optimizers in recurrent neural networks for text classification, c. 1
14. \\tLin, Natural language processing for analyzing disaster recovery trends expressed in large text corpora
15. \\tDharrao, Fractional Krill–Lion algorithm based actor critic neural network for face recognition in real time surveillance videos, Int. J. Comput. Intell. Appl., № 18
16. Patil, Improvement of optical character recognition in images with mixed text through semantic segmentation, J. Sensor Actuator Networks, vol. 11, pp. 63
17. Kumar, Detection of malaria disease using cnn technique with sgd, rmsprop and adam optimizers, Deep Learn. Techniques Biomed. Health Inform., pp. 211-230
18. Shan, Text sentiment analysis method based on CNN-BiGRU in big data environment for social network, Mobile Inf. Syst., 2023

19. Kumari, Performance of Optimizers in Text Summarization for News Articles, Procedia Comput. Sci., № 218, p. 2430.