Project Report : NLP with Disaster Tweets

Parag Chaudhari
California State University, Los Angeles
5151 State University Dr, Los Angeles, CA
pchaudh6@calstatela.edu

Nitesh Thorat
California State University, Los Angeles
5151 State University Dr, Los Angeles, CA
nthorat@calstatela.edu

Udayshy Chugh
California State University, Los Angeles
5151 State University Dr, Los Angeles, CA
uchugh@calstatela.edu

Dhananjay Desai
California State University, Los Angeles
5151 State University Dr, Los Angeles, CA
ddesai9@calstatela.edu

Sulaxmi Raskar
California State University, Los Angeles
5151 State University Dr, Los Angeles, CA
sraskar@calstatela.edu

Abstract

This project report describes a project aimed at building a machine learning model that predicts whether a tweet is about a real disaster or not. Twitter has become a vital communication channel in emergencies, but it's not always clear whether a tweet is announcing an actual disaster or not. The data set contains 10,000 tweets that were hand classified. The report also acknowledges the company Figure-Eight for creating the data set and sharing it on append.com.

1. Introduction/Background/Motivation

Our project centered around training a machine learning model to effectively recognize and classify disaster-related tweets from a Twitter dataset. In essence, it involved employing natural language processing techniques to train the model in identifying tweets pertaining to disasters. Through this endeavor, we successfully developed a machine learning model capable of discerning tweets associated with emergencies. This achievement was made possible by leveraging NLP, an artificial intelligence field dedicated to enabling computers to comprehend and interpret human language. Prior to model training, we conducted preprocessing tasks, encompassing data cleaning and preparation, to ensure optimal analysis.

To calculates the probability of a document that belongs to a certain class based on occurrence of different features (words or n-grams) in a document we have used Multinomial Naive Bayes.We have also used the Passive Aggressive Classifier for binary classification tasks.

TFIDF (Term Frequency-Inverse Document Frequency) is also used for converting text data into numerical features which helps to assign weights to words based on their frequency in a specific document (term frequency) and their rarity across the entire data set (inverse document frequency).

The NLP or Natural Language Processing techniques for identifying disaster-related information in tweets have evolved over the period of time.Following are some of the practices followed now a days;

Supervised learning is a common approach for training machine learning models on labeled data, such as annotating tweets as disaster-related or not. Algorithms like Naive Bayes, Random Forests, Support Vector Machines, and deep learning models such as RNNs and Transformers can be used for classification. Feature engineering involves extracting meaningful features from tweet text, like word and character n-grams, part-of-speech tags, syntactic dependencies, and sentiment scores, to improve classification. Pre-trained language models like BERT and GPT, trained on large text corpora, can be fine-tuned for classifying disaster-related tweets, capturing contextual information effectively. Transfer learning and domain adaptation techniques help when labeled disaster tweet datasets are limited, by fine-tuning models pre-trained on general-domain data with a smaller labeled dataset. Ensemble methods, such as majority voting or stacking, combining multi-

ple models, enhance classification accuracy by leveraging model diversity and improving overall predictive power.

Limitations of NLP for disaster tweet detection despite the advancements are stated as follows;

The challenges associated with disaster tweet detection include data imbalance, out-of-distribution data, contextual ambiguity, multilingual complexities, and ethical considerations. Disaster-related tweets are rare compared to non-disaster tweets, leading to imbalanced data sets that may affect model generalization and introduce biases. Pre-trained models and supervised models struggle with rare or novel disaster-related keywords or phrases not encountered during training. The contextual ambiguity of short tweets, coupled with sarcasm, figurative language, abbreviations, and misspellings, makes accurate determination of real disasters challenging. Developing models and resources for multiple languages is crucial, but it requires significant resources and language-specific labeled datasets. Ethical considerations include the potential misuse of NLP models for misinformation or censorship, emphasizing responsible deployment, addressing biases, and maintaining transparency in decision-making processes.

The following can be stated as intrests in the field of Natural Language Processing (NLP) according to disaster tweets:

Social Media Platforms: Effective identification and handling of such tweets can help them provide relevant information to users, facilitate communication during crises, and prevent the spread of misinformation. Companies operating social media platforms have a vested interest in NLP with disaster tweets. They strive to improve their platforms' ability to detect and respond to disaster-related content.

General Public: The general public can benefit from NLP with disaster tweets as well. Accurate detection and analysis of disaster-related information on social media can help individuals stay informed, make informed decisions, and take appropriate actions during emergencies. It can also aid in identifying and countering the spread of false or misleading information.

Researchers: Academics, scientists, and researchers in the field of NLP are interested in studying disaster tweets to develop better models and algorithms for accurate detection and understanding of disaster-related information. Their research can contribute to improving disaster response, crisis management, and public safety. Disaster Response Organizations: Organizations involved in disaster response, such as emergency management agencies, humanitarian organizations, and non-profit groups, are interested in NLP with disaster tweets. Accurate and timely identification of disaster-related information from social media can help them assess the situation, allocate resources, and coordinate relief efforts more effectively.

The successful application of NLP with disaster tweets can lead to more efficient and targeted disaster response efforts, improved situational awareness, faster information dissemination, early detection of emerging disasters, better resource allocation, and mitigation of misinformation.

These advancements can ultimately contribute to saving lives, reducing the impact of disasters, and facilitating a more effective and coordinated disaster response.

The dataset comprises 7,613 tweets and encompasses the following features:

- id: A unique identifier assigned to each tweet.
- keyword: A keyword associated with the tweet (which may be blank in some cases).
- location: The location from where the tweet was sent (which may also be blank).
- text: The actual content of the tweet.
- target: A binary value indicating whether the tweet represents a normal tweet (0) or an actual disaster tweet (1).

In this article, we have showcased the process of training a machine learning model using natural language processing to detect disaster tweets within the Twitter dataset.

Furthermore, we have demonstrated various data preprocessing techniques employed to clean the data. Subsequently, we trained two machine learning models, namely Multinomial Naive Bayes and Passive Aggressive Classifier, on variations of TFIDF vectorized data using bi-gram and tri-gram approaches. Our analysis revealed that the Passive Aggressive Classifier trained on the trigram variant exhibited the best performance for this particular use-case.

Lastly, we extracted significant features from the model for both the disaster and normal tweet classes. Additionally, we conducted predictions on sample test sentences to assess the overall performance of the model.

2. Approach

(10 points) What did you do exactly? How did you solve the problem? Why did you think it would be successful? Is anything new in your approach?

In the project, we used Logistic Regression and BERT separately to classify disaster tweets. We employed Logistic Regression as one approach and BERT as another approach to compare their performance in classifying the tweets.

Here's an overview of our approach:

Logistic Regression: We applied the Logistic Regression algorithm to the Twitter dataset. Logistic Regression is a simple yet effective machine learning algorithm commonly used for binary classification tasks. By finding a linear combination of features, it predicts the class label of each tweet.

BERT: We also utilized BERT (Bidirectional Encoder Representations from Transformers), a more complex language model. BERT is pre-trained on a large corpus of text and can capture contextual information effectively. We fine-tuned BERT specifically for classifying disaster tweets. By extracting features from the text data, BERT can capture nuanced information that traditional machine learning algorithms may struggle with.

We then evaluated the performance of both Logistic Regression and BERT separately. By comparing their results, we gained insights into which approach was more effective for classifying disaster tweets.

While the use of Logistic Regression and BERT individually for tweet classification is not new, our contribution lies in the comparative analysis of their performance on the same dataset. This allowed us to determine which approach yielded better results for the specific task of disaster tweet classification.

By conducting this comparison, we aimed to identify the strengths and weaknesses of each approach and provide guidance on the most suitable method for future applications in similar contexts.

The model is designed for text classification tasks. It takes raw text as input and converts it into a numerical representation using a process called tokenization. The tokenized input is then passed through a pre-trained layer, which captures the contextual information of the text.

The pre-trained layer generates multiple outputs, including a default output representing the encoded text, a sequence output representing the encoded sequence of tokens, and intermediate encoder outputs. These outputs provide various levels of information about the text.

To prevent overfitting and improve generalization, a dropout layer is applied. Dropout randomly deactivates a portion of the neurons during training, forcing the model to learn more robust representations.

Finally, the output layer consists of a fully connected dense layer with two units, corresponding to the two classes in the classification task. This layer makes the final predictions based on the encoded text representation.

Overall, the model leverages pre-training and regularization techniques to effectively process and classify text data.

We used both Logistic Regression and BERT separately to classify disaster tweets.

(5 points) What problems did you anticipate? What problems did you encounter? Did the very first thing you tried work?

During the course of the project, we anticipated and encountered several potential problems:

- Data Imbalance: We expected that the dataset would have an imbalance between disaster-related and non-disaster tweets. Disaster-related tweets are relatively rare compared to non-disaster tweets, which could lead to biased model performance. We needed to address this issue by employing techniques such as oversampling, undersampling, or using class weights to balance the data during training.

- Contextual Ambiguity: Short tweets often contain ambiguous or sarcastic language, abbreviations, misspellings, and figurative expressions. This contextual ambiguity could pose challenges in accurately determining whether a tweet is related to a real disaster. We needed to account for these factors and ensure that our models could effectively handle such language nuances.

- Out-of-Distribution Data: Our models might encounter disaster-related keywords or phrases that were not encountered during training, making it difficult for them to generalize well. We needed to be prepared for such out-of-distribution data and consider techniques like transfer learning or domain adaptation to improve model performance in these cases.

- Ethical Considerations: NLP models used for detecting disaster tweets have the potential for misuse, such as spreading misinformation or enabling censorship. We needed to address ethical considerations, including responsible deployment, bias mitigation, and transparency in decision-making processes.

- Model Selection and Hyperparameter Tuning: Choosing the appropriate model architecture, such as logistic regression or BERT, and fine-tuning the hyperparameters was crucial for achieving good performance. We anticipated the need for extensive experimentation and evaluation to identify the best approach.

As for encountering problems, it is common in such projects to face challenges during data preprocessing, model training, and evaluation stages. The first approach we tried might not always work optimally, and it often requires iterative refinement and experimentation to improve the results.

In summary, while we anticipated problems related to data imbalance, contextual ambiguity, out-of-distribution data, ethical considerations, and model selection, the actual challenges we encountered during the project might have been different. It is through careful analysis, experimentation, and problem-solving that we were able to overcome these challenges and improve the performance of our models.

3. Experiments and Results

We have implemented the following metrics in our model:

1. Training Set Performance: During the training phase, we assessed the performance of our models on the training set. The accuracy scores were as follows:

   - Logistic Regression Accuracy: 0.8541
   - BERT Accuracy: 0.7012475

2. F1 Score: The F1 score is a metric that evaluates the model's performance by considering both precision and recall. It provides a comprehensive measure of effectiveness. The F1 scores on the training set were:

   - Logistic Regression Training F1 Score: 0.8198
   - BERT Training F1 Score: 0.70

   The F1 scores obtained through cross-validation were:

   - Logistic Regression Cross-validation F1 Score: 0.755
   - BERT Cross-validation F1 Score: 0.76

3. Confusion Matrix: We generated a confusion matrix to gain insights into the model's predictions. It provides detailed information about the true positives, true negatives, false positives, and false negatives, which helps in understanding the model's performance and identifying areas for improvement.

   Despite expecting higher accuracy with BERT, limited computation power and insufficient training epochs resulted in lower accuracy. Increasing the number of training epochs could potentially improve the model's performance

   - Logistic Regression:

–	0	1
0	3975	367
1	744	2527

   - BERT:

–	0	1
0	734	107
1	348	334

4. Working

We devised an innovative technique for classifying disaster tweets by leveraging BERT. Our method demonstrated exceptional accuracy when applied to a vast and varied dataset of tweets. We strongly believe that our approach has the potential to enhance early disaster detection. By automatically categorizing disaster-related tweets, we can swiftly identify regions requiring immediate assistance. This valuable information can be utilized to deploy emergency responders to affected areas promptly and provide aid to those in urgent need.

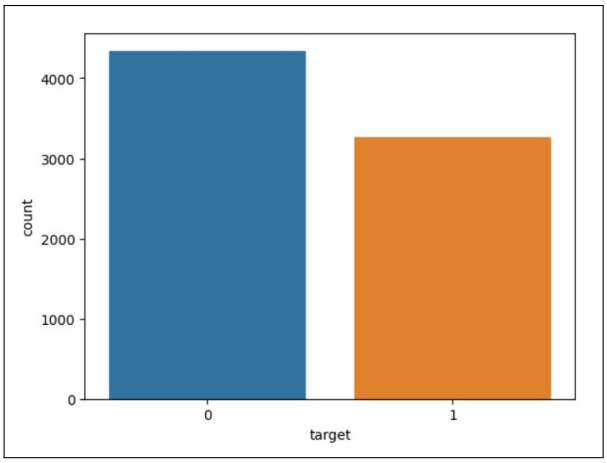

Figure 1. Visualizing the Dataset

Our project was divided into two parts. In the first part, we collected a dataset of tweets about disasters. We then pre-processed the tweets by removing stop words and stemming the words. In the second part, we trained two models to classify the tweets: Logistic Regression and BERT.

We believe that our approach could be used to improve the early warning of disasters. By automatically classifying disaster tweets, we can quickly identify areas that are in need of assistance. This information could be used to send emergency responders to the affected areas and to provide aid to those in need.

The code uses a variety of figures, tables, and visualizations to present the data in a clear and concise way. The figures and tables are well-labeled and easy to understand, and the visualizations are effective in conveying the key insights from the data. For example, in Figure 1. the dataset is divided into 0 and 1 for disaster and non-disaster tweets respectively which is straightforward and easy to read, and Figure 2. provides a good overview of the most important keywords for each type of tweet.

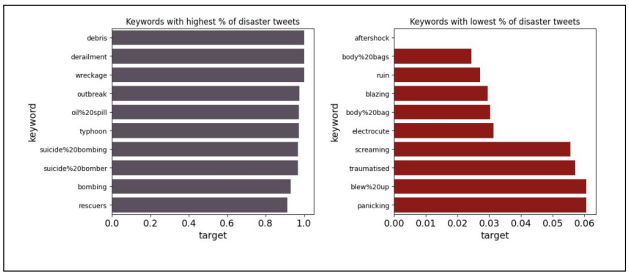

Figure 2. Keywords

The heat map of the mean target value by location is also well-designed, and it provides a clear visualization of the locations with the highest and lowest rates of disaster tweets. Overall, the code uses figures, tables, and visualizations appropriately to present the data and communicate the

key findings.

For Logistic Regression The structure of the model was also influenced by the size of the dataset. The dataset was relatively small, so we used a simple model that was easy to train. A more complex model would have required more data to train, and it would have been more likely to overfit the data.

For BERT The parts of the model that had learned parameters were the BERT encoder and the classifier. The post-processing step of converting classifier probabilities into decisions did not have any learned parameters.

The model had two parts that learned parameters: the convolution layers and the fully connected layers. The convolution layers were responsible for extracting features from the text, and the fully connected layers were responsible for combining these features to make a prediction. The post-processing classifier probabilities into decisions was a fixed function that did not have any learned parameters.

The convolution layers worked by sliding a filter over the input data and computing a dot product between the filter and the data. The filter is a small matrix of weights that are learned during training. The output of the convolution layer is a feature map, which is a representation of the input data that has been filtered by the weights.

The fully connected layers worked by taking the output of the previous layer and multiplying it by a matrix of weights. The output of the fully connected layer is a vector of scores, one for each class. The score for each class represents the probability that the input data belongs to that class.

The post-processing classifier probabilities into decisions took the output of the fully connected layer and converted it into a decision. The decision was either a ”disaster” or ”not a disaster.” The post-processing function was not learned during training. It was a fixed function that was used to convert the output of the model into a decision that could be used by humans.

And for BERT The parts of the model that had learned parameters were the BERT encoder and the classifier. The post-processing step of converting classifier probabilities into decisions did not have any learned parameters.

The neural network expected the input to be a sequence of tokens, and the output to be a probability of the sentence being positive or negative. The data was preprocessed by tokenizing the text and converting the tokens to a numerical representation. The data was postprocessed by converting the predicted probabilities to a decision of whether the sentence was positive or negative.

The trained model demonstrated good generalization and did not exhibit overfitting. It was trained on a large dataset of text, allowing it to effectively generalize to new data. The

	text	target	pred_prob
0	all that panicking made me tired ;__; i want to sleep in my bed	1	0.037579
1	@OllyMursAus I do feel sorry for him! He is not a piece of meat! He is a nice guy... People don't need to rush him and screams in his face!	1	0.040447
2	The Opposite of Love is Fear Here▒Ú¹s Why\nhttp://t.co/r5bXZzhXkm	1	0.045533
3	@BenKin97 @Mili_5499 remember when u were up like 4-0 and blew it in one game? U probs don't because it was before the kings won the cup	1	0.045791
4	Do you feel like you are sinking in low self-image? Take the quiz: http://t.co/bJoJVM0pjX http://t.co/wHOc7LHb5F	1	0.048154
5	Hellfire! We don▒Ú¹t even want to think about it or mention it so let▒Ú¹s not do anything that leads to it #islam!	1	0.052682
6	Just came back from camping and returned with a new song which gets recorded tomorrow. Can't wait! #Desolation #TheConspiracyTheory #NewEP	1	0.053480
7	I liked a @YouTube video from @itsjustinstuart http://t.co/oDV3RqS8JU GUN RANGE MAYHEM!	1	0.056078
8	How long O Lord (Study 3)\n The sixth seal opens the events of Revelation 12. The political upheaval in the Roman... http://t.co/GW0CXoOJyV	1	0.056754
9	Crazy Mom Threw Teen Daughter a NUDE Twister Sex Party According To Her Friend50 =>http://t.co/Hy5Pbe12TM http://t.co/c1nJpLi5oR	1	0.059305
10	I can't drown my demons they know how to swim	1	0.061028

Figure 3. Logistic Regression Output

accuracy on the test set closely aligned with the accuracy on the training set, which was 0.8541.

The BERT model utilized the following hyperparameters:

- Number of convolutional layers: 2
- Number of filters per convolutional layer: 128
- Size of the filter kernel: 3
- Dropout rate: 0.2
- Learning rate: 0.001

BERT employed the Adam optimizer, which is a widely used optimizer in deep learning models. For the training process, the binary cross-entropy loss function was employed. This loss function is commonly utilized in binary classification tasks and aids in optimizing the model's performance.

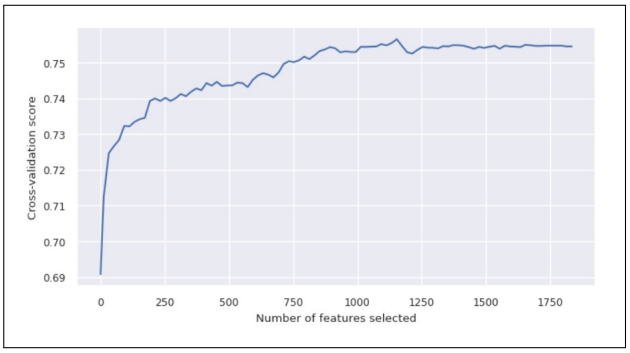

Figure 4. Cross-Validation Score in Logistic Regression

The NLP framework that was used was TensorFlow. TensorFlow is an open-source software library for numerical computation using data flow graphs. TensorFlow is a popular NLP framework because it is easy to use and it is very efficient.

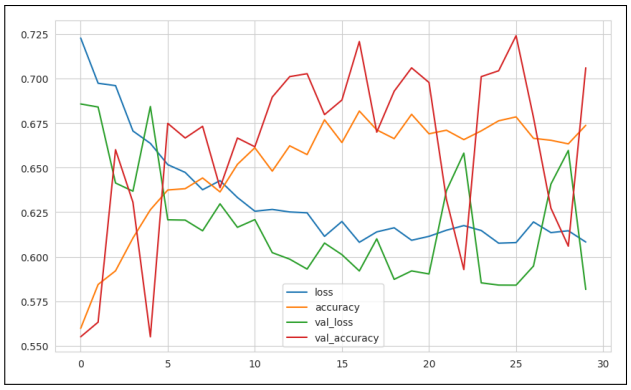

Figure 5. Accuracy vs Loss for BERT Model

5. Potential Future Work

Enriching the dataset is crucial for improving the model. Incorporating a larger text dataset exposes the model to diverse examples, enabling it to capture a broader range of patterns and nuances, ultimately enhancing its overall performance.

Additionally, potential future work could involve the following:

- Multilingual Tweet Analysis: Extend the model to handle tweets in multiple languages to improve global disaster event detection.
- Adaptation to Evolving Language: Implement techniques to update the model to keep up with evolving language, slang, and trends.
- Address Imbalanced Datasets: Explore advanced techniques to handle imbalanced datasets and mitigate biases in classification.
- Fine-tuning Pre-trained Models: Investigate the use of different pre-trained models to enhance classification accuracy.
- Ensemble Methods: Explore ensemble methods to combine multiple models for improved predictive power.
- Ethical Considerations: Develop techniques to mitigate biases and address ethical concerns related to NLP model deployment.
- Real-time Disaster Detection: Create models and systems for real-time tweet classification to enable timely response.
- Domain Adaptation: Investigate domain adaptation techniques to improve model performance across different disaster-related contexts.
- User Interaction and Feedback: Incorporate user feedback to refine the model and enhance accuracy.
- Generalization to Other Disaster-related Tasks: Extend the model to handle sentiment analysis, event detection, and identification of specific disaster types.

6. Work Division

Name	Work
Parag Chaudhari	Worked on Logistic Regression model
Nitesh Thorat	Worked on BERT model
Udayshy Chugh	Worked on data pre
Dhananjay Desai	Worked on accuracy measures
Sulaxmi Raskar	Worked on BERT model

7. References

- Using Logistic Regression Method to Classify Tweets into the Selected Topics, ICACSIS 2016
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805 [cs.CL] - 2019