

Natural Language Processing with Disaster Tweet

TEAM 04

Shrinivas Bhusannavar, Yashasvi Kotra, Monica Lokare, Karthik Nimmagadda

Harivardhana Naga Naidu Polireddi, Ravjot Singh

Department of Applied Data Science, San Jose State University

DATA 255 - Deep Learning Technologies

Dr. Simon Shim

December 08, 2024

Natural Language Processing with Disaster Tweets

1. Introduction

The increasing prevalence of social media sites such as Twitter has been a game-changer for disaster reporting and response. However, it is quite tricky to identify tweets about real disasters out of the figurative and non-relevant ones. This project applies Natural Language Processing techniques to classify tweets as disaster-related or not using state-of-the-art models like LSTM, BERT, Advanced BERT, and RoBERTa. The textual data will come from a Kaggle competition to determine whether the tweet really talks about a disaster or not. It will then classify the tweets accordingly, which is very crucial for many applications such as emergency response, disaster management, and real-time alert systems. These models utilize both word embeddings, such as GloVe, and transformer-based architectures, including RoBERTa, which employs optimized training strategies like dynamic masking and robust pre-training, to achieve high classification performance. By leveraging these advanced techniques, this project aims to deliver a reliable and scalable solution for real-world disaster tweet classification.

1.1 Background

Social media have emerged recently as a critical real-time information source in the time of disasters and emergencies. Social networking sites like Twitter allow individuals to report updates, call for help, and give first-hand insights, making it indispensable during crisis management. The data volume produced during those times often includes irrelevant or non-disaster-related content, which makes filtering and prioritizing useful information very challenging. With the increase in unstructured data, especially in textual format, manual processing of such data is not practical anymore. This demands the design and development of automated systems capable of classifying tweets as either related or unrelated to disasters. Such systems will help authorities and organizations concentrate their efforts on actionable insights, thereby speeding up the process of disaster response with added efficiency.

1.2 Problem Statement

The objective of this project is to develop a robust and scalable model for classifying tweets as either disaster-related or non-disaster-related. This classification is critical for enabling faster response times during emergencies and allocating resources efficiently. Specifically, the project aims to address the following challenges:

1. Extracting meaningful patterns from noisy, unstructured tweet data.
2. Capturing contextual information to distinguish between disaster-related and irrelevant tweets.
3. Leveraging state-of-the-art machine learning techniques to maximize accuracy and minimize misclassification. By solving this problem, the project seeks to provide a reliable and automated solution for processing large-scale social media data during disasters.

1.3 Significance

The success of this project will have wide-ranging effects on disaster management systems. This project aids in automating the classification of tweets.

Real-Time Monitoring: The quick identification of relevant disaster-related tweets enables emergency responders to trace the ongoing events as they are happening.

Efficient Resource Allocation: The filtering of actionable information helps authorities to provide resources to the most urgent locations.

Improved Decision-Making: With accurate and reliable tweet classification, organizations can better prioritize interventions and reduce response times.

Community Safety: Indirectly, this project enhances the safety and well-being of the affected communities by streamlining the flow of critical information. The project aims to bridge the gap between unstructured social media data and actionable insights, supporting a more efficient and effective disaster response framework.

1.4 Scope

Project utilizes advanced deep learning techniques and tools for classifying disaster-related tweets.

Data Source & Description

- **Source:** [Kaggle: NLP Getting Started](#)
- **Data Files:**
 - train.csv: Labeled tweets (text, keyword, location, target)
 - test.csv: Unlabeled tweets for final submission
- **Target Variable:** target (1 = disaster-related, 0 = not related)

The major components involved are as follows:

Data Preprocessing: Cleaning the raw tweet data by removing URLs, special characters, and noise. Tokenization and padding sequences for maintaining the same input length.

Feature Engineering: Pre-trained GloVe embeddings, which encode semantic relationships between words. BERT tokenizer, which tokenizes words with context.

Model Selection

1. LSTM (Long Short-Term Memory):

LSTM, a recurrent neural network, serves as the baseline model. It is designed to capture sequential dependencies in text, leveraging its memory cells and gating mechanisms to process input data over time. Pre-trained GloVe embeddings are used to initialize word representations, enabling the model to utilize semantic knowledge from large-scale corpora.

2. BERT (Bidirectional Encoder Representations from Transformers):

BERT revolutionizes NLP by introducing a transformer-based architecture that processes text bidirectionally. By capturing both preceding and succeeding context, BERT significantly improves understanding of sentence structure and meaning. Fine-tuning the pre-trained bert-base-uncased model enables task-specific performance optimization.

3. Advanced BERT:

Advanced BERT builds on the capabilities of BERT by incorporating larger model variants (e.g., bert-large-uncased) and optimizing hyperparameters like learning rates and

batch sizes. These enhancements result in improved precision, recall, and F1-score, making the model highly reliable for disaster tweet classification.

4. **RoBERTa (Robustly Optimized BERT):**

RoBERTa further refines the BERT architecture by employing dynamic masking, removing the next sentence prediction task, and pre-training on larger datasets. These optimizations enhance robustness and generalization, achieving state-of-the-art results for disaster classification tasks.

Evaluation

To ensure model performance meets real-world standards, several evaluation metrics are employed, including:

- **Accuracy:** Measures the proportion of correctly classified tweets.
- **Precision:** Indicates the proportion of true positives among all positive predictions.
- **Recall:** Measures the ability to identify true positives from all actual positives.
- **F1 Score:** Balances precision and recall providing a comprehensive performance measure.

2. Literature Review/Related Work

2.1 Literature Survey

NLP has transformed those segments that demand accuracy and speed, such as disaster management and healthcare. In support, Lakshmi Narayana U.'s "Detecting Disaster Tweets using a Natural Language Processing Technique" (2023) identified NLP's role in disaster-related tweet classification for real-time disaster management. The study also highlighted how machine learning techniques extracted important information from social media data to assist in responding to disasters.

The Biom MRA model is an example of the potential that NLP has in dealing with such complex medical datasets in biomedical applications. Biom MRA, trained on data from the Mina Foundation and PetCentral, uses RAG to deliver precise multilingual medical information. Its integration of the L-Chain framework for language modeling and Free Cab library for processing PDFs enables effective analysis of long medical texts. Biom MRA thus performs efficiently on resource-constrained devices, optimized through quantization and model merging, and proves to be valuable in a wide range of healthcare professional contexts. Biom MRA is an open-source model that has reported the best results on medical question-answering tasks, inviting continuous improvement and wider application. This shows how NLP is going to transform decision-making from disaster management to biomedical data analysis.

The field of sentiment analysis, generally in non-English languages, has seen significant advancements through the application of machine learning techniques. Altawaier and Tiun (2016) conducted a study, "Comparison of Machine Learning Approaches on Arabic Twitter Sentiment Analysis," to understand the effectiveness of various machine learning methods for analyzing sentiment in Arabic tweets. The study has also pointed to the linguistic complexities of Arabic: it is a dialectally rich language with rich morphology, which complicates sentiment analysis models' performance (Altawaier & Tiun, 2016).

In their study, Altawaier and Tiun implemented several machine learning methods- such as support vector machines, naïve Bayes, and k-nearest neighbors-comparing each against the Arabic sentiment classification tasks. Support vector machines always

outperform other models in terms of accuracy and robustness on handling Arabic text, as the outcome has shown. The work also outlines the importance of preprocessing, stemming, and tokenization for enhancing sentiment classification.

The findings reveal the important role machine learning can play in the areas of linguistics challenges and insights extracted from social media data. By giving a comparative study of the various approaches, the work will be useful in informing future work in Arabic sentiment analysis and has a larger implication for the potential of machine learning in multilingual sentiment classification.

The book "Natural Language Processing with Python" by Bird, Klein, and Loper (2009) has been a source book for both practitioners and researchers in the field of NLP. It introduces NLP concepts and techniques in detail, with the strong emphasis on practical applications using the Python programming language. This book now uses the Natural Language Toolkit, a comprehensive Python library, to demonstrate how to perform tasks such as tokenization, stemming, part-of-speech tagging, and parsing (Bird et al., 2009).

One of the main strengths of the book is how it is equally accessible for both the beginner and advanced user. It covers the basics, but also more advanced topics such as building language models and using machine learning algorithms for NLP. It contains hands-on exercises and practical examples that will help the readers bridge the gap between theory and application. Also, the book emphasizes how important preprocessing is in NLP pipelines. It provides detailed explanations of linguistic structures for better understanding of complex text processing tasks. It is a very important resource in democratizing NLP by making sophisticated tools and techniques accessible to an audience far beyond the leading research

teams. With a focus on open-source tools and hands-on implementation, it is one of the cornerstones for both teaching and applying NLP in academia and industry alike.

The study "Introduction to NTL with Disaster Tweets" by Elhariri (2021) analyzes tweets about disaster events using some Natural Language Processing (NLP) techniques. It highlights how NLP can be leveraged to classify and extract critical information from large volumes of unstructured social media data in real time. The research focuses on using NLP Text Labeling (NTL) techniques to identify actionable insights, such as emergency needs, damage reports, and resource availability, during disasters (Elhariri, 2021).

The preprocessing steps of tokenization, stemming, and stopword removal, among others, are highly stressed in the work of Elhariri to ensure that text data is of good quality. It also delves into the role of supervised machine learning algorithms in categorizing tweets, demonstrating their potential in disaster response efforts. This study provides a practical framework for deploying NLP tools to manage the information overload that often characterizes crisis situations and hence get quicker and more efficient responses.

This paper uses social media analysis integrated with NLP methodologies to develop a scalable solution for disaster management in real time. The research underlines the crucial role of NLP in enhancing situational awareness and decision-making, hence being an asset for researchers and practitioners within the domain of disaster informatics.

The paper "RoBERTa: A Robustly Optimized BERT Pretraining Approach" by Liu et al. (2019) introduced some important enhancements in the domain of NLP, building upon the earlier model of BERT with robust optimization. RoBERTa further refines the

pretraining process of BERT by removing the NSP objective and training on larger datasets with longer sequences, leading to improved model performance on a variety of NLP tasks.

Key contributions of the present research: extending training time, using much larger datasets, and introducing dynamic masking strategies in pretraining. It illustrates that such advances yield state-of-the-art performance on established NLP benchmark tasks including GLUE, SQuAD, and RACE datasets. Compared to BERT, RoBERTa adopts a simpler yet more efficient approach, showing that careful optimization can significantly outperform complex methods that introduce unnecessary complexities.

This work underlines the potentiality of pretraining strategies in improving the generalization capabilities of language models. By focusing on robust optimization rather than architectural changes, RoBERTa set the benchmark for future research in pretraining methods and NLP applications.

The study "A Comparative Study of Chatbots and Humans" by Mittal et al. (2016) deals with the ever-changing face of chatbots in interacting with humans and compares it with human performances in effective communication. This study addresses the capability and limitation of the chatbot to perceive and respond to natural language queries while focusing on their efficiency, scalability, and handling of repetitive tasks.

The authors have underlined that advances in NLP and machine learning have allowed chatbots to be at the forefront of customer service, health, and education. Nevertheless, this study also recognizes serious drawbacks: chatbots cannot understand subtlety in language, emotions, and context like humans do. Such a comparative analysis would point to the fact that while a chatbot does a great job with the automation of

mundane and predictable tasks, human intervention becomes quite important when the going gets complex and requires loads of empathy and deeper understanding.

This research underlines the complementary nature of chatbots and humans, calling for hybrid systems that combine the strengths of both. It provides very valuable insights into the design and deployment of conversational agents, making it a foundational reference for future advancements in chatbot technologies.

The tutorial "NLTK Sentiment Analysis: Text Mining Analysis in Python" by Navlani (2019) offers a gentle introduction to performing sentiment analysis using the Natural Language Toolkit, NLTK, in Python. Much importance is attached to the preprocessing of text data through tokenization, stopwords removal, and stemming in order to enhance the performance of the sentiment classification models. By exploiting the inbuilt functionalities of NLTK, the tutorial shows how one can analyze text data and draw meaningful insights from it effectively.

Navlani shows the application of sentiment analysis on customer feedback, product reviews, and social media analysis. The tutorial covers a variety of techniques in text preprocessing, building a basic model for sentiment classification, and its performance evaluation on labeled datasets. In general, examples here act as a hands-on guide for the text mining and NLP beginner.

This work indicates the flexibility of NLTK when it comes to sentiment analysis, as well as Python's usability in performing text analytics. It remains a useful tool for novices who have a desire to apply NLP techniques in real-world scenarios.

The paper "Sentiment Analysis or Opinion Mining: A Review" by Saad and Saberi (2017) provides a comprehensive overview of sentiment analysis techniques and their

applications across various domains. The study explores the evolution of sentiment analysis, focusing on the methodologies, tools, and challenges associated with extracting subjective information from unstructured text data. It highlights the importance of sentiment analysis in understanding customer opinions, market trends, and public sentiment (Saad & Saberi, 2017).

Sentiment analyses could be performed at the document level, sentence level, or aspect level; the relevance of each to a certain context is highlighted. Among others, machine learning approaches, both supervised and unsupervised methods, are also reviewed that have been in vogue lately toward raising the accuracy of the classification model. Handling sarcasm, negation, and sentiments based on contexts remains some of the challenges that call for additional studies and innovations.

This work acts as a foundational reference to understand the scope and limitations of sentiment analysis. It contributes to the development of more robust and versatile systems for sentiment analysis applicable to various industries by providing insights into current practices and future directions.

The paper "A Proposed Solution for Sentiment Analysis on Tweets to Extract Emotions from Ambiguous Statements" by Shukla et al. (2015) addresses one of the most challenging aspects of sentiment analysis: extracting emotions from ambiguous and context-dependent statements commonly found in tweets. The study focuses on overcoming the inherent limitations of traditional sentiment analysis methods when applied to short, informal, and noisy text data, such as social media content. The authors have proposed an improved methodology that combines preprocessing techniques and state-of-the-art machine learning models to enhance the accuracy of sentiment

classification (Shukla et al., 2015).

The proposed solution involves a robust preprocessing pipeline that comprises tokenization, stopword removal, stemming, and part-of-speech tagging to prepare the raw tweet data for analysis. The authors have also highlighted the importance of feature extraction methodologies comprising n-grams, TF-IDF for the capture of relevant context from tweets. Further, the study has explored the application of supervised machine learning algorithms, including Support Vector Machines and Decision Trees, to classify emotions pertaining to happiness, sadness, anger, and sarcasm.

The main contribution of this paper is that it focuses on handling ambiguous language, sarcasm, and mixed sentiments, which are more frequent in tweets and get usually misclassified by the traditional systems. By handling these challenges, the authors show an improvement in emotion detection and sentiment classification. The findings of the study have practical implications for industries relying on social media analytics, including marketing, customer service, and public opinion monitoring.

The work reported herein constitutes a substantial step in the development of more nuanced sentiment analysis systems, which can take care of social media data. This will be a great, much-needed detailed framework for researchers and practitioners working on emotion detection in ambiguous textual data.

The paper "A Survey on Feature Level Sentiment Analysis" by Joshi and Itkat 2014 discusses the various advancements and methodologies in the sentiment analysis domain, focusing most of their discussion on the feature-level approach. Unlike document or sentence-level sentiment analysis, feature-level sentiment analysis drills down to a finer level of attributes or aspects that are mentioned in text. This granularity is particularly

useful in applications such as product reviews, where customers often express varying sentiments about different features of the same product (Joshi & Itkat, 2014).

The authors present the state of the art in a few different active areas, reporting challenges in feature extraction, sentiment polarity determination, and aggregation. The main tasks of NLP necessary to identify relevant features in SA are part-of-speech tagging, dependency parsing, and named entity recognition. Machine learning algorithms are considered one of the most critical building blocks to enhance accuracy and scalability for feature-level models in SA.

It also points out some key challenges in handling implicit features, sarcasm, and the influence of context on sentiment, emphasizing the complexity of accurately extracting and classifying sentiments at the feature level. The paper concludes by suggesting future directions for research, such as integrating deeper semantic analysis and leveraging hybrid approaches that combine rule-based and machine learning techniques.

The paper surveys the emerging domain of sentiment analysis and the insight into how feature-level analysis can enhance decision-making in applications related to e-commerce, marketing, and customer relationship management.

The paper "Incorporating Sentiment Prior Knowledge for Weakly Supervised Sentiment Analysis" by He (2012) explores how to incorporate prior sentiment knowledge into weakly supervised models for sentiment analysis in order to further improve their accuracy and reliability. Unlike fully supervised methods, which rely on substantial labeled data, the principle of weak supervision seeks to minimize this dependency by informing model training with external sources of sentiment knowledge, such as lexicons and pre-defined rules describing sentiment expressions.

His approach illustrates how the inclusion of prior sentiment knowledge can overcome challenges either in text with sparse labels or ambiguous sentiments. This study has highlighted the effectiveness of using sentiment lexicons, polarity scores, and semantic similarity measures to augment feature extraction and classification processes. Furthermore, the research proposes algorithms for dynamic updating of prior knowledge while the model is learning to ensure adaptability to different contexts and datasets.

The paper further discusses the application of this methodology to multilingual sentiment analysis and domain-specific tasks, where labeled data is usually scarce. Combining weak supervision with sentiment priors, the model achieves performance competitive with fully supervised models while keeping computational efficiency.

This work highlights the importance of external sentiment resources in improving weakly supervised learning methods. It forms a very important reference to researchers who want to come up with scalable and adaptable sentiment analysis models that can be efficient even when data is limited.

The paper "Scene Classification Using Support Vector Machines with LDA" by Veeranjanyulu et al. (2014) introduces a new scene classification approach by combining Support Vector Machines with Linear Discriminant Analysis. The study tries to enhance the performance of scene classification models by capitalizing on the strengths of LDA for dimensionality reduction and that of the SVM for robust classification.

The authors have proposed the use of LDA for discriminative feature extraction from high-dimensional scene data so that computational efficiency and noise in the feature space are improved. These features will be fed into the SVM classifier, which is well-known for its generalization capability, to carry out the scene categorization tasks. This

combination enables the model to handle changes in light conditions, perspectives, and scene complexities-an issue generally faced in scene classification.

Experimental results shown in this work have demonstrated that the LDA-SVM approach outperforms a single classifier or any other conventional methods in classification performance. Further, the authors have discussed the scalability of their proposed approach toward real-world applications in image retrieval, surveillance, and autonomous systems.

This work emphasizes hybrid methodologies in computer vision and will go a long way toward the advancement of techniques related to scene classification. This is an LDA-SVM combination, which balances the efficiency-accuracy trade-off well, hence being an extremely useful framework for several practical applications.

Elhariri (2021), within the article "Introduction to NLP with Disaster Tweets," gives an enlightening approach toward the use of NLP techniques in the analyses of tweets during disasters. The article identifies the importance of making use of social media data for real-time disaster management through the classification of tweets, which helps in pinpointing relevant information such as emergency needs and damage reports.

The work underlines some of the preprocessing steps necessary for raw text data preparation, such as tokenization, stemming, and stopword removal. It further deliberates on the use of some supervised learning models, like Logistic Regression and Support Vector Machines, in classifying tweets. Elhariri also introduces advanced NLP techniques, including feature extraction with Term Frequency-Inverse Document Frequency, to enhance the model's ability to spot meaningful patterns within the data.

One of the high points of this article is the fact that it dwells on the practicality of

deploying NLP models in disaster response. By automating the classification of disaster-related tweets, the approach ensures faster and more efficient information dissemination to aid emergency services in resource allocation and decision-making.

This research underlines the importance of NLP in social media analytics, especially in situations where fast and accurate processing of information is required. Therefore, it is a very useful source for both researchers and practitioners in disaster informatics.

Sharma (2021), in the article "BERT for Identifying Disasters from Tweets," explores the use of Bidirectional Encoder Representations from Transformers (BERT) for analyzing and classifying disaster-related tweets. The study highlights the importance of leveraging social media platforms for real-time disaster response by identifying tweets that provide actionable information, such as resource needs and damage reports (Sharma, 2021). It talks about the advantages of using BERT because it is able to understand the context and long-range dependencies of the text, which is important for accurate classification. Sharma goes into the pre-processing steps that need to be taken, including tokenization and input formatting required by the BERT model, and fine-tuning of the model to adapt it to disaster tweet classification.

The results of the analysis depict that BERT outperforms other traditional machine learning models, such as Logistic Regression and Support Vector Machines (SVM), on grounds of higher accuracy and generalizing well on unseen data. These results also provide practical insights regarding challenges in working with an imbalanced dataset and, hence, strategies that might help to mitigate the challenges by oversampling or through weighted loss functions.

This work underlines the transformative potential of transformer-based architectures

like BERT in social media analytics, particularly for tasks requiring rapid and precise classification. This may be a valuable resource for researchers and practitioners working on disaster informatics and real-time decision-making systems.

Sharma and Aakanksha present a comparative study of sentiment analysis using rule-based and Support Vector Machine in the paper "A Comparative Study of Sentiment Analysis Using Rule-Based and Support Vector Machine". This paper deeply analyzes the different techniques for sentiment classification, with much emphasis on strengths and weaknesses of rule-based methods in comparison to Support Vector Machines. Sentiment analysis involves the extraction of subjective information from text data, an important process in applications related to customer feedback and social media monitoring, as identified by Sharma & Aakanksha (2014).

The rule-based approach depends on pre-defined linguistic rules and sentiment lexicons for classification. Though simple and interpretable, it usually has poor performance for complex sentences, sarcasm, and context-dependent sentiments. On the other hand, SVM is a supervising machine learning algorithm that does an excellent job of handling high-dimensional feature space and non-linear relationships in text data. In this study, these methods were compared across datasets, showing SVM to outperform rule-based methods in terms of accuracy and scalability consistently.

However, the authors point out that "the performance of SVM heavily depends on feature selection and preprocessing techniques such as stemming and TF-IDF vectorization." Meanwhile, rule-based methods remain useful for applications where high interpretability and minimal computational resources are required. The concluding remark is that the paper's choice between these two approaches depends on the peculiar demands

of the task concerning labeled data availability and textual complexity.

This research highlights the trade-offs between rule-based and machine learning-based sentiment analysis techniques, offering valuable insights for researchers and practitioners who want to improve user engagement in ebusiness and e-learning platforms.

Krishnakumar 2021, in the article "Natural Language Processing with Disaster Tweets: Part 1," presents a complete overview of applying NLP techniques to classify disaster-related tweets. This study is intended to take full advantage of text classification in separating those tweets that give information that may be acted upon during disasters and those that do not. This is important because correct classification could mean the difference in real-time disaster response and resource allocation. The article highlights how preprocessing, which entails text cleaning, tokenization, and lemmatization of raw tweet data, is essential in getting this data into an analyzable format. Krishnakumar introduces different methods of feature extraction, like TF-IDF and word embeddings, which transform text data into a format suitable for machine learning models. Furthermore, some results for the application of conventional machine learning models, such as Logistic Regression and Support Vector Machines-SVM, and deep learning in tweet classification are compared in the study.

The most important part of this work is the fact that it is an application-oriented piece, showing clearly how these techniques can be applied in real-world scenarios to process volumes of social media data efficiently. The article shows the potential of NLP in disaster management by making better decisions faster and creating improved situational awareness during emergencies.

The study hereby presents valuable insights on how NLP can be used for social

media analytics in the context of disaster informatics and, therefore, stands to serve as a useful guide to both researchers and practitioners in this field.

Alhammadi (2022), the author of the paper "Using Machine Learning in Disaster Tweets Classification," has investigated machine learning techniques for the classification of disaster-related tweets. The paper pinpoints that social media data are increasingly important in the management of disasters since timely identification of relevant tweets can help in the better allocation of resources and response (Alhammadi, 2022).

The paper reviews those supervised machine learning algorithms, which are widely used for this purpose, like Logistic Regression, Support Vector Machines (SVM), and Random Forest. Alhammadi stresses the crucial role preprocessing techniques, such as tokenization, stemming, and stopword removal, play in cleaning and furnishing meaningful input to those models. Another study examines the adequacy of several feature extraction methodologies, such as bag-of-words and Term Frequency-Inverse Document Frequency (TF-IDF), for the efficient capturing of patterns in the text.

Among the findings, preprocessing with advanced machine learning algorithms considerably raises the classification accuracy. This paper discusses challenges dealing with an imbalanced dataset for disaster tweet classification and suggests methods to alleviate the issues through oversampling and cost-sensitive learning techniques.

It epitomizes how machine learning contributes value toward disaster informatics by helping classify tweets for real-time information analysis. The research contribution is huge regarding the practice of machine learning on social media in different disasters.

Chanda (2021), in the paper "Efficacy of BERT Embeddings on Predicting Disaster from Twitter Data," analyses the performance of Bidirectional Encoder Representations

from Transformers embeddings concerning the classification of disaster-related tweets.

This study showed that there is a capability of BERT embeddings to learn deeper contextual features from text data and, as such, it can be useful in the proper discrimination of disaster-related tweets from unrelated ones.

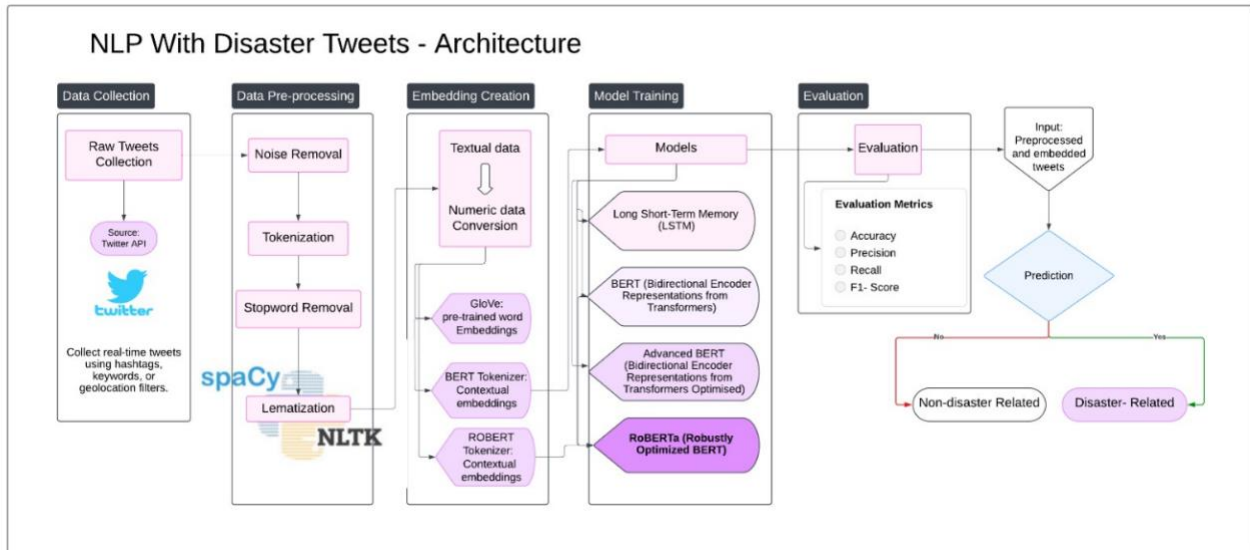
This research elucidates the benefits of using BERT over traditional machine learning and feature-based methods. The model leverages pre-trained embeddings to capture semantic relationships and contextual meaning that are usually lost in simpler bag-of-words or TF-IDF approaches. Chanda evaluates the performance of BERT embeddings on a labeled dataset of tweets, showing a great improvement in classification accuracy, precision, and recall compared to traditional models such as Logistic Regression and Support Vector Machines.

This research further explores the challenges of operating with short and noisy text data, which is common on social media platforms like Twitter. By fine-tuning BERT for disaster tweet classification, the study gives practical insights into adapting transformer-based models for domain-specific tasks. Chanda emphasizes the importance of preprocessing and balanced datasets to optimize model performance.

This work emphasizes how advanced NLP techniques have transformed the face of social media analytics, especially in disaster informatics, whereby timely and efficient tweet classification can greatly improve the effectiveness of emergency response efforts.

3. Project Architecture

Figure 1: *Architecture Diagram*



The architecture of the "NLP with Disaster Tweets" project will be in such a way that it will classify tweets into disaster-related and non-disaster-related categories. It goes through a structured pipeline starting with data collection, going through data preprocessing, embedding creation, model training, and evaluation, and ending with prediction and post-processing.

Each step of the pipeline provides a crucial contribution to make the classification system highly effective and accurate, hence targeted for real-time disaster monitoring and response.

3.1 Data Collection

The first step in the pipeline involves the collection of tweets via the Twitter API. This allows the collection of real-time tweets by filtering through hashtags, keywords, or geolocation parameters. These raw tweets form the backbone of the entire process and capture information that might be indicative of disaster-related activity in its raw, unfiltered form. The project collects continuously updated data from Twitter to

keep the model relevant and responsive to events that are still unfolding.

This step is quite important because it lays the basis for the next steps in analysis.

3.2 Data Preprocessing

The raw tweets may contain a lot of noise and inconsistencies that may weaken the model's performance. At this stage, the cleaning and preparation of data head the preprocessing. Noise removal includes the removal of useless elements such as hashtags, mentions, URLs, or special characters. Further, cleaned text was tokenized, which refers to breaking down the preprocessed text into smaller units or words for further analysis. In other words, all stop words (commonly used words that do not carry meaningful information, such as "and" "the") are removed in order to retain only the core. Finally, lemmatization (using tools like NLTK and spaCy) reduces words to their base forms, ensuring consistency and improving the quality of model input. This ensures that the preprocessed data is clean, structured, and ready for embedding.

3.3 Embedding Creation

In this stage, the cleaned and preprocessed textual data is transformed into numerical vectors that machine learning models can interpret. The project employs several embedding techniques to achieve this, starting with **GloVe** (Global Vectors for Word Representation), which provides dense vector representations of words based on their semantic meaning. Additionally, **BERT Tokenizer** is used to create contextual embeddings, where the meaning of words is determined by their surrounding context. For more advanced and robust representation, **RoBERTa** embeddings are employed, offering optimized contextual embeddings that are particularly

effective for nuanced understanding of tweets. This transformation from text to numerical data is critical, as it bridges the gap between natural language and computational models.

3.4 Model Training

The numerical embeddings are fed into multiple models for training, each designed to handle the unique challenges of tweet classification. The first model, **Long Short-Term Memory (LSTM)**, excels in capturing sequential patterns in text data but may struggle with contextual nuances. To address this, the project utilizes **BERT (Bidirectional Encoder Representations from Transformers)**, which analyzes text in both directions to provide deep contextual understanding. Advanced versions of BERT, fine-tuned for specific tasks, are also employed to improve accuracy. Finally, **RoBERTa (Robustly Optimized BERT)**, a cutting-edge transformer model, is used for its ability to handle intricate details in tweets with exceptional performance. The models are trained on labeled datasets containing disaster-related and non-disaster-related tweets. Evaluation metrics such as **accuracy**, **precision**, **recall**, and **F1-score** are used to assess their performance, ensuring that the most effective model is selected for deployment.

3.5 Prediction

The trained models are then used for classifying the incoming tweets. The input to this stage is the preprocessed and embedded tweets, passed through the trained models to produce a binary classification of tweets as disaster-related or not. This stage represents the core functionality of the system, providing immediate and actionable results. By properly distinguishing between relevant and irrelevant tweets, this step ensures that only the important information is flagged for further analysis.

4. Data Exploration

4.1 Dataset Overview

- **Training Dataset:** 7,613 samples with id, keyword, location, text, and target columns.
- **Testing Dataset:** 3,263 samples with similar features, excluding target.

Figure 2: *Dataset Overview*

```
Training Data Sample:
  id keyword location text \
0   1   NaN     NaN Our Deeds are the Reason of this #earthquake M...
1   4   NaN     NaN           Forest fire near La Ronge Sask. Canada
2   5   NaN     NaN All residents asked to 'shelter in place' are ...
3   6   NaN     NaN 13,000 people receive #wildfires evacuation or...
4   7   NaN     NaN Just got sent this photo from Ruby #Alaska as ...

  target
0        1
1        1
2        1
3        1
4        1

Testing Data Sample:
  id keyword location text
0   0   NaN     NaN           Just happened a terrible car crash
1   2   NaN     NaN Heard about #earthquake is different cities, s...
2   3   NaN     NaN there is a forest fire at spot pond, geese are...
3   9   NaN     NaN           Apocalypse lighting. #Spokane #wildfires
4  11   NaN     NaN Typhoon Soudelor kills 28 in China and Taiwan
```

4.2 Null Value Analysis

A significant number of missing values were observed in the keyword and location columns, as visualized in the Null Value bar chart. Text and target columns were fully populated.

Figure 3: *Null Value Analysis*



Target Class Distribution

The dataset showed a slight imbalance, with non-disaster tweets (57%) outnumbering disaster-related tweets (43%).

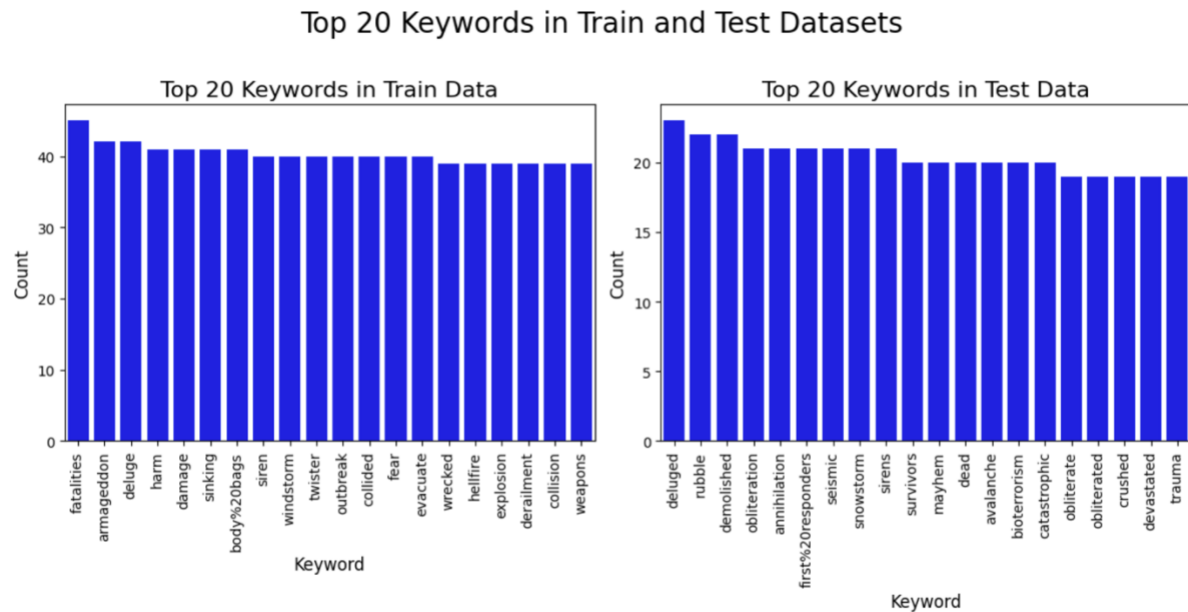
Figure 4: *Target Class Distribution*



4.4 Keyword and Location Analysis

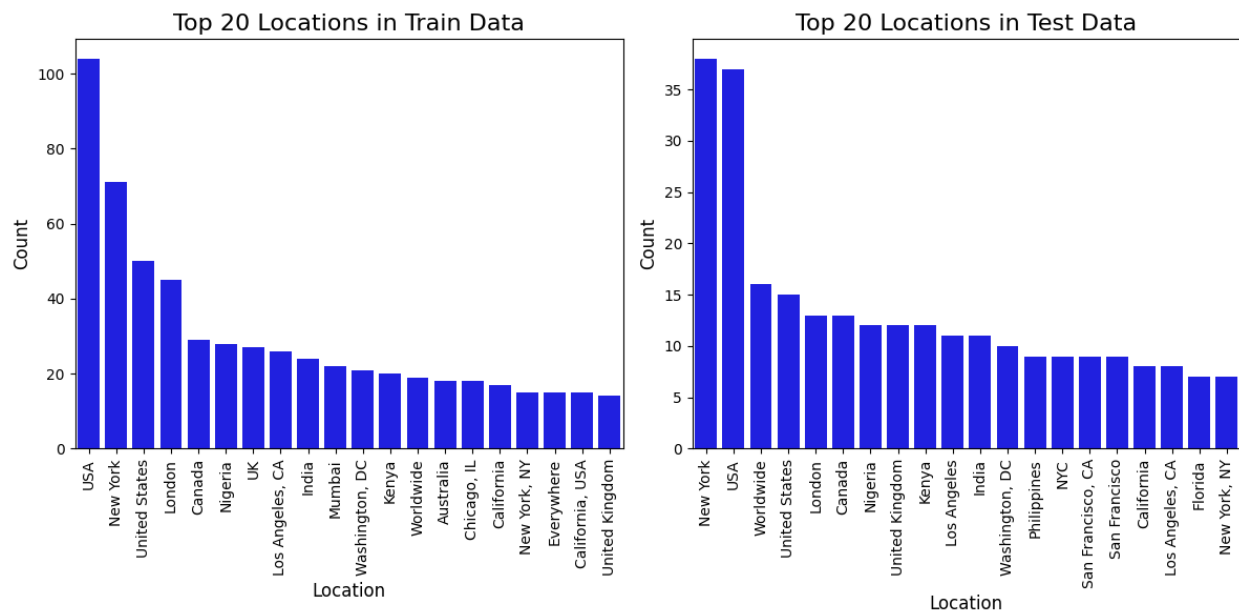
Top keywords like "fatalities" and "armageddon" were more prevalent in disaster tweets, while common non-disaster terms included "youtube" and "content policy."

Figure 5: *Keyword Analysis*



Locations such as "USA," "New York," and "London" appeared frequently across both datasets.

Figure 6: *Location Analysis*



5. Text Preprocessing

The raw tweet data often contains noise in the form of URLs, emojis, special characters, and other irrelevant elements that can negatively impact the model's ability to learn meaningful patterns.

5.1 Data cleaning processes

The following steps were undertaken to clean the text data:

- **Removal of URLs:** Links and URLs were removed using regular expressions (`http\S+`) to eliminate unnecessary distractions.
- **Handling Mentions:** Twitter handles (e.g., `@username`) were removed as they are not informative for disaster classification.
- **Removal of Emojis:** Emojis were stripped from the text to reduce noise using libraries such as `emoji` or custom regex patterns.
- **Stripping HTML Tags:** Any HTML-like content (e.g., `
`, `&`) was removed to focus on plain text.
- **Punctuation Removal:** All punctuation marks (e.g., `!`, `?`, `#`) were removed to simplify the tokenization process.
- **Lowercasing:** All text was converted to lowercase to ensure consistency during tokenization and to avoid treating the same word in different cases as separate tokens.
- **Stopword Removal (Optional):** Commonly used words such as "and", "the," and "is" were retained in some cases, as they could provide context, especially for BERT.

This step helps in the text data to be clear, concise, and free of irrelevant content, making it ready for downstream processing.

Figure 6: Cleaned Text

	id	keyword	location	text	target	label_names	text_clean
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	disaster	Our Deeds are the Reason of this earthquake Ma...
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	disaster	Forest fire near La Ronge Sask Canada
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	disaster	All residents asked to shelter in place are be...
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	disaster	13000 people receive wildfires evacuation orde...
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1	disaster	Just got sent this photo from Ruby Alaska as s...
...
7608	10869	NaN	NaN	Two giant cranes holding a bridge collapse int...	1	disaster	Two giant cranes holding a bridge collapse int...
7609	10870	NaN	NaN	@aria_ahrary @TheTawniest The out of control w...	1	disaster	ariaahrary TheTawniest The out of control wild...
7610	10871	NaN	NaN	M1.94 [01:04 UTC]?5km S of Volcano Hawaii. htt...	1	disaster	M194 0104 UTC5km S of Volcano Hawaii
7611	10872	NaN	NaN	Police investigating after an e-bike collided ...	1	disaster	Police investigating after an ebike collided w...
7612	10873	NaN	NaN	The Latest: More Homes Razed by Northern Calif...	1	disaster	The Latest More Homes Razed by Northern Califo...

7613 rows x 7 columns

5.2 Word Embeddings

GloVe (Global Vectors for Word Representation) was used to convert words into dense numerical vectors, capturing semantic meanings and relationships between words. Pretrained GloVe embeddings (glove.6B.100d.txt) were loaded, providing 100-dimensional vectors for each word in the vocabulary. For words that were not present in the GloVe vocabulary, for those random embeddings were generated with a normal distribution. A custom embedding matrix was created where each row corresponds to a word in the vocabulary, ensuring compatibility with the LSTM model. The embeddings allowed the LSTM model to leverage semantic relationships between words, improving its ability to classify tweets based on context.

5.3 Tokenization Process

Tokenization is the process of splitting text into smaller units, such as words or subwords, which are then mapped to numeric representations.

Different tokenization strategies were employed for the LSTM and BERT models:

- **Tokenization for LSTM:**

- Each tweet was split into individual words using a custom tokenizer or Python's `split()` method.
- A vocabulary was created from the training body, where each word was assigned a unique index.
- Unknown words (not in the vocabulary) were mapped to a special <UNK> token to handle out-of-vocabulary terms.

- **Tokenization for BERT:**

- BERT uses a WordPiece tokenizer, which splits words into subwords to reduce the size of the vocabulary and handle rare words.
- Special tokens, such as [CLS] (classification token) and [SEP] (separator token), were added to each sequence.
- The tokenizer ensured that longer words were broken into meaningful subwords ("classification" into "class ##ification"), allowing BERT to understand complex vocabulary.

The tokenized output for both models was transformed into sequences of numeric IDs corresponding to the words or subwords in the tweet.

5.4 Sequence Padding

Since models require inputs of consistent length, padding was applied to ensure that all tokenized sequences had the same number of elements. Padding helps in batching data for training and evaluation.

- **LSTM Padding:**

- Sequences were padded with zeros (<PAD> token) up to a fixed length (e.g., 50 tokens).
- For tweets longer than the fixed length, truncation was applied to keep only the most relevant words.

- **BERT Padding:**

- Padding was performed to the maximum sequence length observed in the dataset or a predefined limit (128 tokens).
- Attention masks were generated alongside the padded sequences. These masks indicate which tokens are actual data (value 1) and which are padded tokens (value 0), ensuring that the model focuses only on the relevant parts of the sequence.

Data Example:

- Original tweet: "Hurricane approaching the coast!"
- Tokenized (LSTM): [34, 98, 124, 5, 56]
- Padded (LSTM): [34, 98, 124, 5, 56, 0, 0, 0, 0, 0] (if max length = 10)
- Tokenized (BERT): [101, 9382, 15342, 1996, 3671, 102]
- Padded (BERT): [101, 9382, 15342, 1996, 3671, 102, 0, 0, 0, 0] (if max length = 10)

Next, reduced words to their base forms using SpaCy. These preprocessing techniques ensured that the text data was clean, standardized, and ready for input into the LSTM, BERT, and Advanced BERT models, maximizing their ability to learn and generalize from the data.

Figure 7: Cleaned Dataset After Text Cleaning

id	keyword	location	text	target	label_names	text_clean	tokenized	lower	stopwords_removed	lemmatized
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	disaster	Our Deeds are the Reason of this earthquake Ma...	[Our, Deeds, are, the, Reason, of, this, earth...	['our', 'deeds', 'are', 'the', 'reason', 'of', '...', 'the', '...', 'the...]	deed reason earthquake allah forgive
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	disaster	Forest fire near La Ronge Sask Canada	[Forest, fire, near, La, Ronge, Sask, Canada]	['forest', 'fire', 'near', 'la', 'ronge', 'sas...]	forest fire near la ronge sask canada
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	disaster	All residents asked to shelter in place are be...	[All, residents, asked, to, shelter, in, place...	['all', 'residents', 'asked', 'to', 'shelter', '...', 'shelter', '...', 'shelter', '...', 'shelter', '...]	resident ask shelter place notify officer evac...
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	disaster	13000 people receive wildfires evacuation orde...	[13000, people, receive, wildfires, evacuation...	['13000', 'people', 'receive', 'wildfires', 'e...]	13000 people receive wildfire evacuation order...
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1	disaster	Just got sent this photo from Ruby Alaska as s...	[Just, got, sent, this, photo, from, Ruby, Ala...	['just', 'got', 'sent', 'this', 'photo', 'from...]	get send photo ruby alaska smoke wildfire pour...

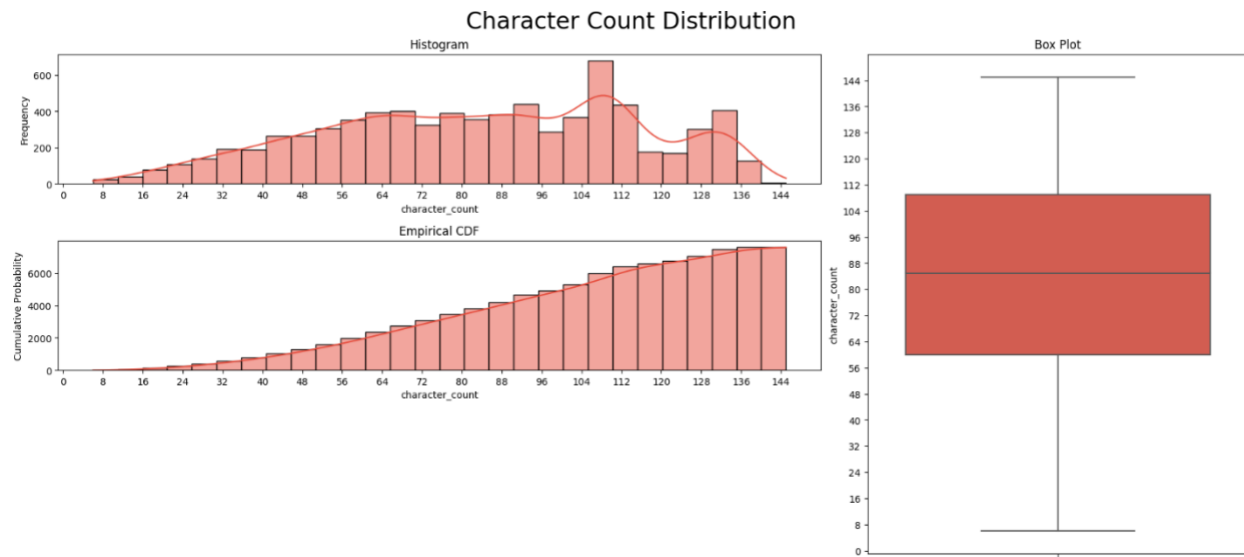
6. Visualization

The visualizations highlight various characteristics of tweets from the dataset, focusing on differentiating between disaster and non-disaster tweets through quantitative and qualitative metrics. Below, each plot is discussed along with its significance in understanding the data.

6.1 Character Count Distribution

The histogram and CDF reveal that the character count for non-disaster tweets is slightly shorter on average compared to disaster-related tweets. This difference indicates potential brevity in casual or non-critical tweets.

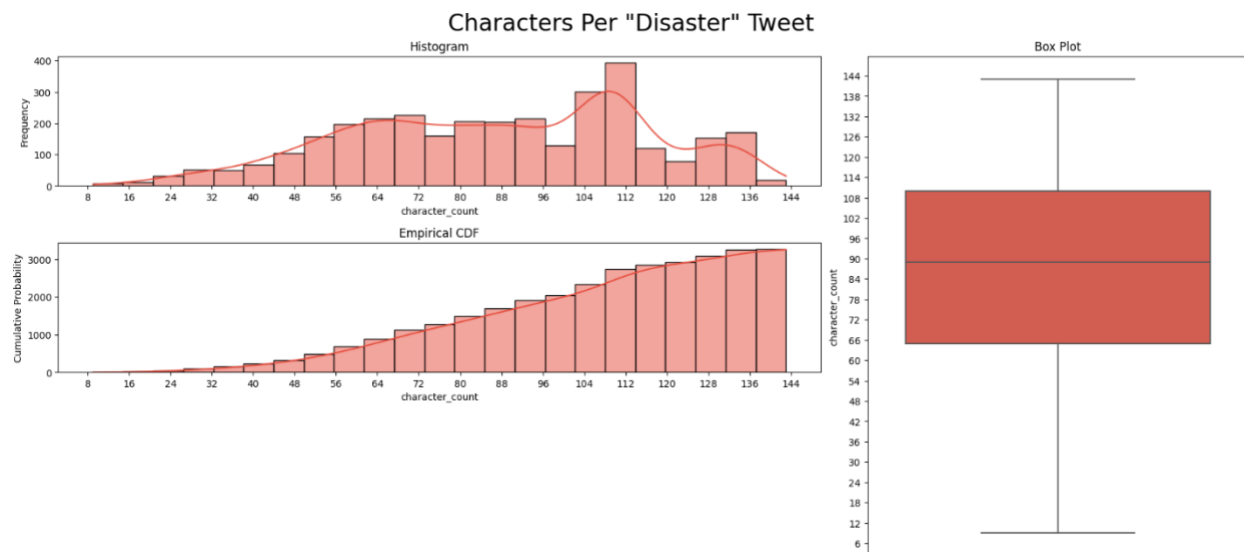
Figure 8: Character Count Distribution



6.2 Disaster Tweets

Tweets related to disasters exhibit a wider range of character counts, likely due to the inclusion of detailed reports or updates. The box plot shows that disaster tweets tend to have higher median character counts, with more variability and outliers.

Figure 9: *Characters Per Disaster Tweet*

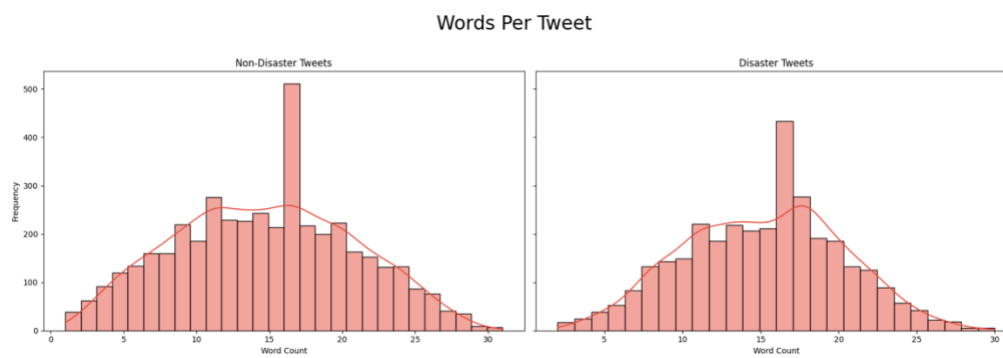


Character count is a useful feature for distinguishing between tweet types, as disaster tweets often convey more information.

6.3 Words Per Tweet

Side-by-side histograms comparing word counts in disaster and non-disaster tweets. Non-disaster tweets exhibit a peak at around 15 words, indicating concise communication. Disaster tweets have a similar peak but exhibit a slightly broader distribution, reflecting varied styles of reporting. Word count offers a complementary feature to character count for identifying tweet types.

Figure 10: *Words Per Tweet*

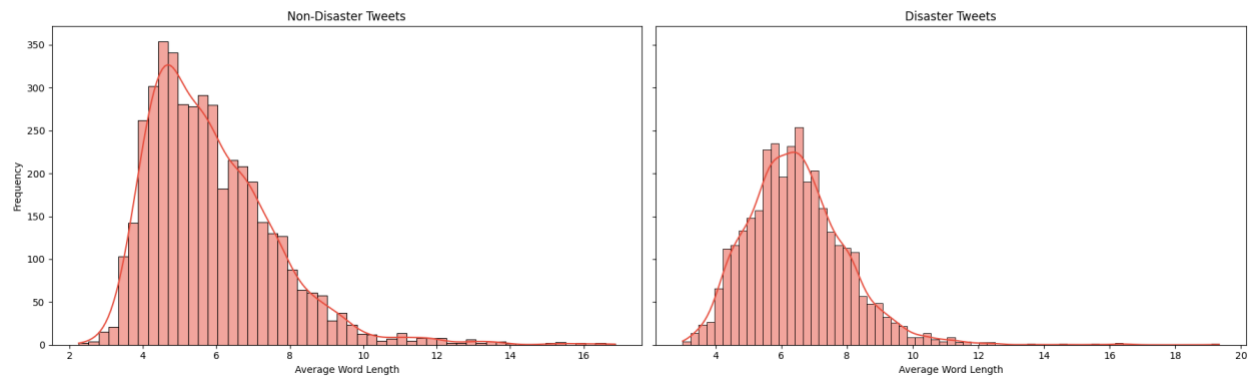


6.4 Mean Word Length

Side-by-side histograms comparing mean word lengths in disaster and non-disaster tweets. The average word length in disaster tweets is marginally higher than in non-disaster tweets, possibly reflecting the use of specific or technical terms. Non-disaster tweets show a higher frequency of shorter words, likely due to casual language. Mean word length can act as a proxy for linguistic complexity, aiding in tweet classification.

Figure 11: *Mean Word Length*

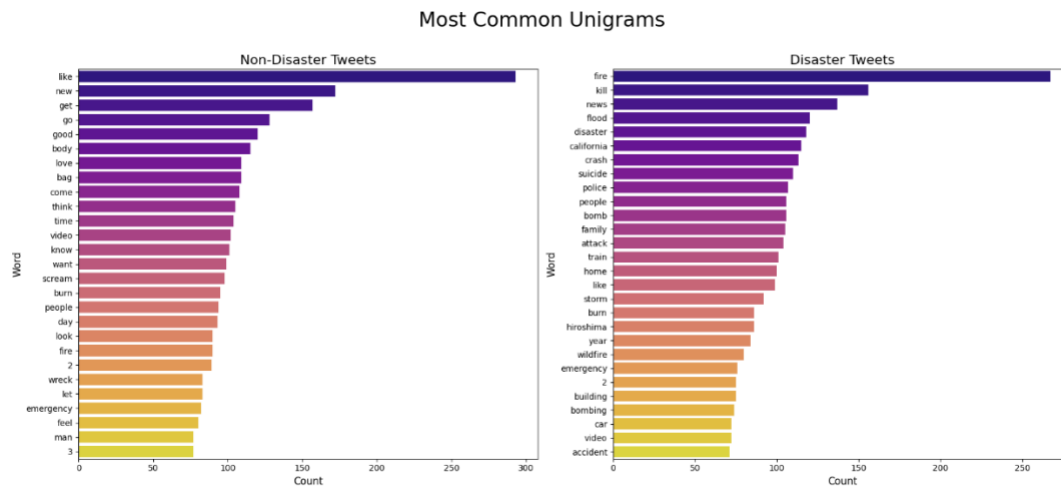
Mean Word Lengths



6.5 Unigrams

Disaster tweets prominently feature words like "fire," "flood," and "disaster," reflecting urgency and context. Non-disaster tweets frequently include casual terms like "like" and "new."

Figure 12: *Most Common Unigram*

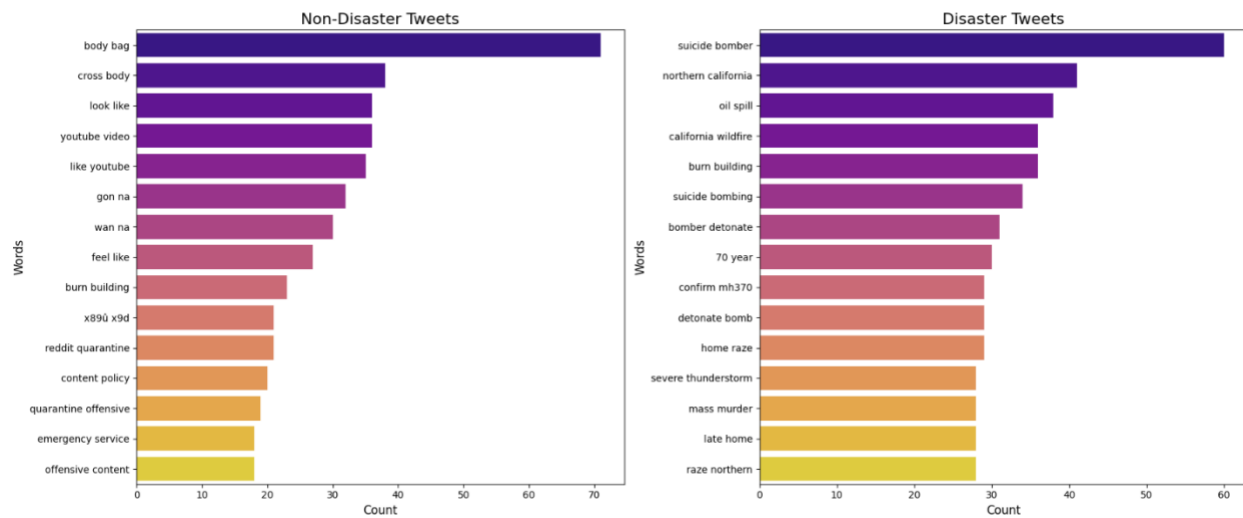


6.6 Bigrams

Common bigrams in disaster tweets include "suicide bomber" and "oil spill," while non-disaster tweets feature phrases like "body bag" and "look like."

Figure 13: *Most Common Bigrams*

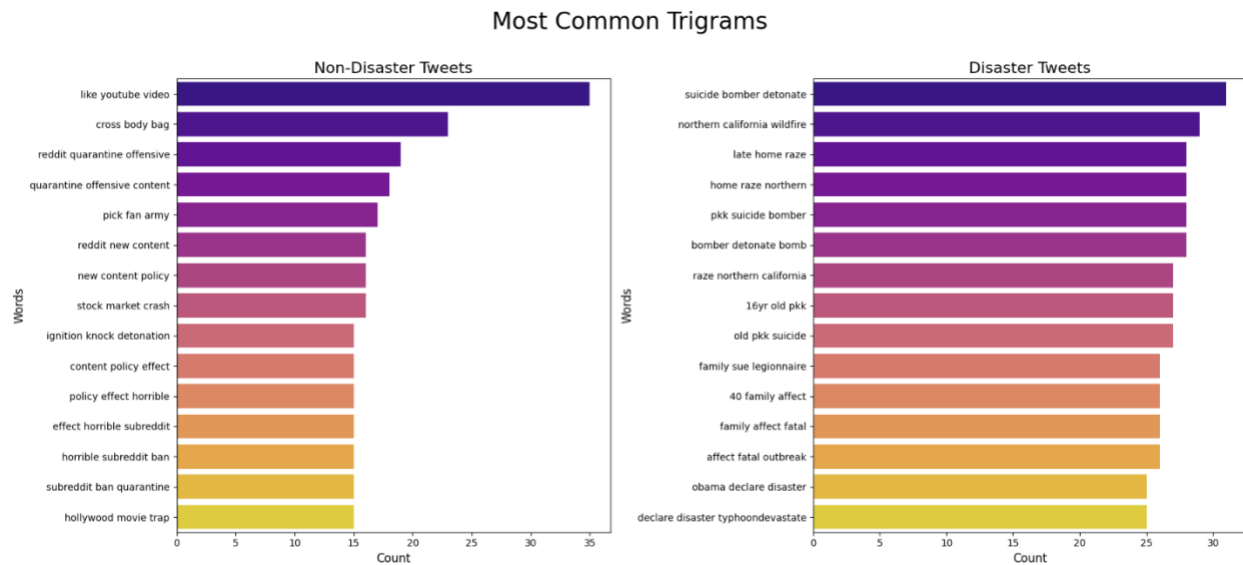
Most Common Bigrams



6.7 Trigrams

Disaster-related trigrams often describe specific events (e.g., "suicide bomber detonate"), whereas non-disaster trigrams show colloquial expressions (e.g., "like youtube video").

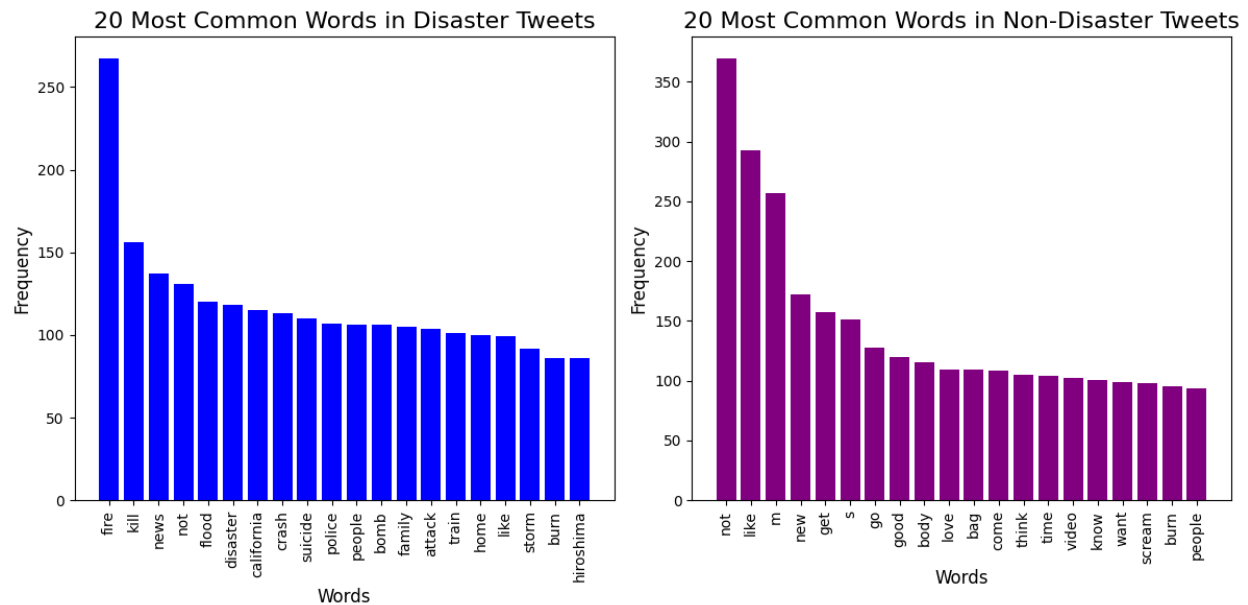
Figure 14: *Most Common Trigrams*



N-grams capture contextual nuances, essential for feature engineering and downstream model training.

Frequency analysis to identify the most common words in disaster and non-disaster tweets. N-gram analysis to capture key phrases and their frequency.

Figure 15: 20 Most Common Words in Disaster and Non-Disaster Tweets



6.8 Word Cloud Visualizations

Word clouds for disaster and non-disaster tweets. Disaster tweets dominated by keywords like "fire," "storm," and "death," emphasizing the gravity of reported incidents. Non-Disaster tweets casual and positive words like "love," "like," and "new" are prevalent. Word clouds provide an intuitive summary of tweet themes, useful for quick data exploration.

Figure 16: Word Cloud for Disaster and Non- Disaster Tweets.

- **Context Understanding:** Many disaster-related tweets use sarcasm or ambiguous phrases, requiring robust semantic understanding.

Formulation:

- **Input:** A tweet in raw text form.
- **Output:** Binary label (1 for disaster, 0 for non-disaster).
- **Objective:** Maximize classification performance using metrics such as **F1 score, precision, recall, and accuracy.**

The problem requires preprocessing text, extracting meaningful features, and selecting appropriate deep learning models for classification.

7. Modeling

This project reflects a journey through four distinct yet interconnected models: Long Short-Term Memory (LSTM), BERT (Bidirectional Encoder Representations from Transformers), Advanced BERT, and RoBERTa (Robustly Optimized BERT). Each step marked a significant leap in understanding, implementing, and overcoming the challenges of working with textual data.

7.1 Model 1: LSTM: The Beginning of Sequential Modeling

The journey began with Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN) that resolves the vanishing gradient problem typical of earlier RNNs. LSTM's gated architecture allows it to selectively remember or forget information across a sequence, making it ideal for tasks like sentiment analysis and text classification. It offered a solid foundation to understand how sequential data could be modeled effectively.

Technical Overview:

The LSTM implementation relied on embedding pre-trained word vectors using models like GloVe. The LSTM layer processed these embeddings to capture dependencies in the input text. The architecture included a dropout layer to prevent overfitting, a fully connected layer to map the final hidden states to output labels, and an optional bidirectional setup to capture dependencies from both directions of the sequence.

Architecture Details:

- **Embedding Layer:** Pre-trained GloVe embeddings (100-dimensional) were used to represent words semantically.
- **LSTM Layer:** A bidirectional LSTM with two layers captured both forward and backward dependencies in the input sequence, with a hidden state size of 128.
- **Fully Connected Layer:** A dense layer mapped the final hidden state to the output class for binary classification tasks.
- **Dropout Layer:** A dropout rate of 0.5 was applied to prevent overfitting.

The model was initialized with hyperparameters like embedding dimensions, hidden state size, the number of LSTM layers, and dropout rate. An Adam optimizer and a learning rate scheduler were used to stabilize the training process. Despite these efforts, it became clear that LSTM's sequential nature inherently limited its ability to model relationships between distant words effectively.

Hyperparameters:

- Hidden Dimension: 128
- Embedding Dimension: 100
- Number of LSTM Layers: 2

- Bidirectional: Yes
- Dropout Rate: 0.5
- Batch Size: 32
- Learning Rate: 0.001
- Optimizer: Adam
- Loss Function: Binary Cross-Entropy with Logits

LSTM effectively captured long-term dependencies in text, leveraging pre-trained embeddings to enhance semantic understanding. However, its reliance on sequential processing limited its performance when compared to modern transformer-based models.

Challenges:

While LSTM performed well for shorter sequences, it struggled with global context, often leading to a trade-off between accuracy and computational efficiency.

Training LSTM on long text sequences was resource-intensive and required careful tuning to mitigate overfitting.

7.2 Model 2: BERT: (A Shift in NLP)

Recognizing the limitations of LSTM in handling global context, the project transitioned to BERT, a model that transformed NLP by introducing bidirectional attention mechanisms. Unlike LSTM, which processes sequences step-by-step, BERT looks at the entire sequence at once, leveraging Transformers' self-attention mechanism to weigh the importance of each word in the context of all others.

Technical Overview:

BERT's architecture comprises multiple Transformer encoders, each consisting of self-attention layers and feed-forward networks. Input embeddings are a combination of token embeddings, segment embeddings, and positional embeddings, ensuring that the model has a sense of word order and sentence segmentation. The fine-tuning phase added task-specific dense layers to predict outputs for classification or regression tasks.

Architecture Details:

- **Base Model:** The pre-trained bert-base-uncased model included 12 layers, 768 hidden units, 12 attention heads, and ~110M parameters.
- **Classification Head:** A single linear layer processed the [CLS] token to predict binary labels.
- **Tokenization:** Input text was tokenized into subwords using the BERT tokenizer, with [CLS] and [SEP] special tokens. Inputs were padded or truncated to a fixed length of 128 tokens.

Using Hugging Face Transformers simplified the fine-tuning process. Pre-trained weights from bert-base-uncased were loaded, and the model was fine-tuned on downstream tasks with hyperparameters like learning rate, batch size, and number of epochs. Gradient accumulation was employed to simulate larger batch sizes on limited GPU resources.

Hyperparameters:

- Batch Size: 16

- Epochs: 3
- Learning Rate: 2×10^{-5}
- Optimizer: AdamW
- Gradient Clipping: Norm capped at 1.0
- Sequence Length: 128 tokens

BERT's bidirectional context understanding enabled it to outperform LSTM across tasks.

However, its high computational requirements and resource-heavy architecture posed challenges during fine-tuning.

Challenges:

Switching to BERT introduced new challenges, particularly with computational requirements.

The quadratic scaling of memory with input sequence length made it necessary to limit sequences to a maximum token length of 128 or 512.

Another challenge was understanding and adapting to WordPiece tokenization, which breaks words into subwords, complicating the preprocessing pipeline.

7.3 Model 3: Advanced BERT (Tailoring for Specific Needs)

While BERT provided exceptional performance on general tasks, its architecture allowed for domain-specific fine-tuning, leading to what can be called Advanced BERT models. These models are customized by pre-training or fine-tuning on datasets tailored to specific industries or problems.

Technical Overview:

Advanced BERT models took BERT's capabilities further by employing larger architectures, task-specific fine-tuning, and optimized hyperparameters. This iteration focused on improving precision, recall, and F1 scores across complex datasets

Architecture Details:

- **Base Model:** The pre-trained Bert-large-uncased model expanded to 24 layers, 1024 hidden units, 16 attention heads, and ~340M parameters.
- **Classification Head:** A linear layer used the pooled output from the [CLS] token for binary classification.
- **Tokenization:** Similar to BERT, but with inputs padded/truncated to a length of 84 tokens for task-specific optimization.

The fine-tuning involved optimizing domain-specific tasks by adjusting parameters such as the learning rate (typically lower for pre-trained layers) and employing techniques like gradual unfreezing, where earlier layers were unfrozen progressively to allow fine-grained tuning.

Hyperparameters:

- Batch Size: 32
- Epochs: 3
- Learning Rate: 6×10^{-6}
- Optimizer: AdamW
- Gradient Clipping: Norm capped at 1.0

- Sequence Length: 84 tokens

Fine-tuning Advanced BERT with larger models and optimized hyperparameters improved its performance, but this came at the cost of increased computational and memory demands.

Challenges:

Handling the large model size of Advanced BERT required GPU resources with higher memory capacities. Moreover, fine-tuning required more time and careful tuning of parameters to achieve optimal results.

Training on domain-specific corpora required, high-quality datasets. Additionally, fine-tuning on tasks with limited labeled data risked overfitting, necessitating careful regularization and dropout configurations.

7.4 Model 4: RoBERTa: (Robustly Optimized BERT)

The final step in this journey was adopting RoBERTa, which builds on BERT's architecture but introduces key refinements. RoBERTa removes the next sentence prediction objective used in BERT, instead focusing solely on masked language modeling with dynamic masking.

Technical Overview:

RoBERTa retains the same core architecture as BERT but includes optimizations in training. It dynamically masks tokens during training, ensuring that the model learns to generalize better across varying input patterns

RoBERTa refined BERT by removing the next sentence prediction task, using dynamic masking

during pre-training, and training on larger datasets with more extensive computational resources. These changes made RoBERTa more robust and better generalized.

Architecture Details:

- **Base Model:** The roberta-base model retained the 12-layer transformer architecture but introduced dynamic masking and larger pre-training corpora.
- **Classification Head:** The pooled [CLS] token was passed through a dense layer for binary classification tasks.
- **Tokenization:** RoBERTa's tokenizer dynamically masked inputs during pre-training, improving generalization.

Using roberta-base from the Hugging Face library, the model was fine-tuned for text classification tasks. Dynamic masking and longer pre-training periods were leveraged to boost robustness. The use of gradient clipping helped prevent exploding gradients during optimization, ensuring stable training.

Hyperparameters:

- Batch Size: 16
- Epochs: 3
- Learning Rate: 1×10^{-5}
- Optimizer: AdamW
- Gradient Clipping: Norm capped at 1.0
- Sequence Length: 128 tokens

RoBERTa delivered state-of-the-art results through its optimized training regimen and dynamic masking approach. However, its reliance on large datasets and extensive computational resources made it less accessible for smaller-scale projects.

Challenges:

Dynamic masking during fine-tuning added complexity to preprocessing pipelines. Achieving optimal performance required significant experimentation with batch sizes, learning rates, and other hyperparameters.

While RoBERTa delivered improved performance, the computational resources required for fine-tuning remained a major hurdle. The model's sensitivity to hyperparameters, such as learning rate and batch size, demanded extensive experimentation to achieve optimal performance.

The journey of model development began with traditional methods like Long Short-Term Memory (LSTM) networks, which provided a solid foundation for handling sequential data. However, as the complexity of language tasks increased, there was a natural progression toward more sophisticated and context-aware models like BERT (Bidirectional Encoder Representations from Transformers), Advanced BERT, and eventually RoBERTa. Each step in this evolution not only addressed the limitations of the previous models but also embraced new challenges and opportunities in natural language understanding.

8. Evaluation & Performance Comparisons

Evaluating the performance of NLP models is a critical step in understanding their effectiveness and reliability. For this project, the evaluation primarily focused on **accuracy**, **precision**, **recall**, and the **F1-score**, with the **F1-score** being the primary metric due to its balanced consideration of both false positives and false negatives.

The results from LSTM, BERT, Advanced BERT, and RoBERTa models provide valuable insights into their strengths and limitations, demonstrating a clear progression in performance.

8.1 Metrics Overview

The following evaluation metrics were used:

1. **Accuracy:**

Measures the proportion of correctly classified samples across all predictions. While intuitive, accuracy can be misleading on imbalanced datasets.

Figure 16: *Accuracy Formula*

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

2. Precision:

Indicates the proportion of true positives among all predicted positives. It is critical in applications where false positives carry a higher cost.

Figure 17: *Precision Formula*

$$precision = \frac{TP}{TP + FP}$$

3. Recall (Sensitivity):

Measures the proportion of true positives among all actual positives. Recall is crucial in tasks where false negatives are costly.

Figure 18: *Recall Formula*

$$recall = \frac{TP}{TP + FN}$$

4. F1-Score:

The harmonic mean of precision and recall, providing a balanced evaluation that penalizes extreme imbalances in either metric.

Figure 19: *F1 Score Formula*

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

8.2 Model-Specific Analysis

The models were evaluated using these metrics, and their respective performance is summarized in the table below:

Model	Accuracy	Precision	Recall	F1 Score
LSTM	0.80	0.76	0.79	0.77
BERT	0.84	0.82	0.79	0.81
Advanced BERT	0.84	0.85	0.84	0.83
RoBERTa	0.84	0.85	0.84	0.80

LSTM

- **Strengths:** Captured long-term sequential dependencies, making it a good baseline model.
- **Weaknesses:** Struggled with nuanced contextual understanding, especially for complex sentences or distant dependencies. This resulted in relatively lower precision (0.76) and F1-score (0.77).
- **Analysis:** LSTM performed adequately for simpler tasks but showed limitations in bidirectional and contextual comprehension.

BERT

- **Strengths:** Significant improvements in precision (0.82) and F1-score (0.81) due to its ability to capture bidirectional context and utilize large pre-trained embeddings.
- **Weaknesses:** While recall remained steady at 0.79, the model's computational intensity required significant resources for training.

- **Analysis:** BERT demonstrated a balanced improvement across metrics, benefiting from its transformer-based architecture.

Advanced BERT

- **Strengths:** Delivered balanced precision (0.85) and recall (0.84), achieving an F1-score of 0.83. Fine-tuned hyperparameters and optimized learning schedules contributed to its improved performance.
- **Weaknesses:** High resource demands due to the larger model size.
- **Analysis:** Advanced BERT proved to be a reliable model, achieving superior performance through task-specific optimizations.

RoBERTa

- **Strengths:** Achieved the highest precision (0.85) and maintained strong recall (0.84), making it highly effective for tasks requiring robust generalization. Its training on larger datasets and dynamic masking contributed to its robustness.
- **Weaknesses:** The model slightly lagged in F1-score (0.80), as precision gains were not perfectly balanced with recall.
- **Analysis:** RoBERTa showcased its ability to generalize well across datasets, outperforming other models in precision but requiring extensive computational resources.

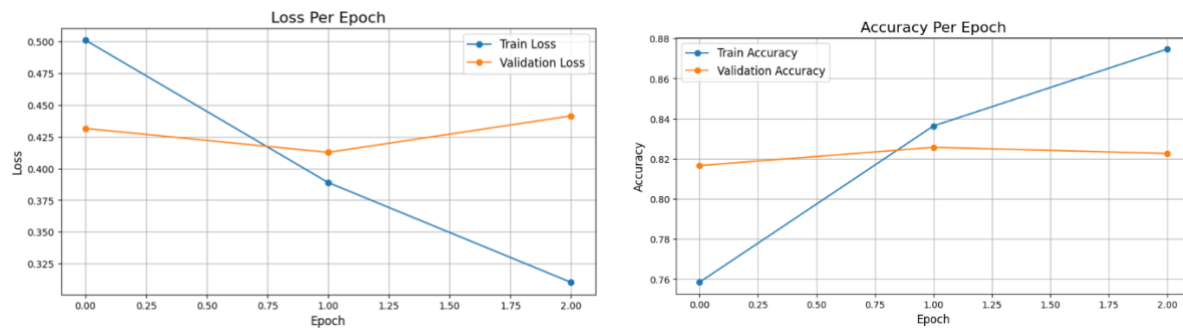
9. Visualizing the Metrics

The evaluation metrics can be further illustrated to highlight the differences in performance:

9.1 LSTM Model Training:

Training is conducted for **3 epochs** with a batch size of 32. Binary cross-entropy loss is used as the objective function. Adam optimizer with a learning rate of 0.001 facilitates parameter updates.

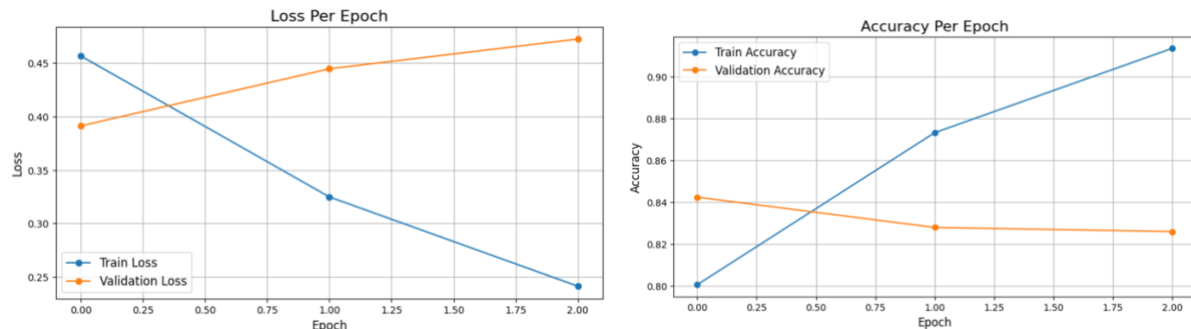
Figure 20: *LSTM Model Training*



9.2 BERT Model Training:

The pretrained BERT model is fine-tuned with a learning rate of $2e-5$ for **3 epochs**. AdamW optimizer updates parameters, and a linear learning rate scheduler is used for gradual decay. Gradient clipping (norm capped at 1.0) prevents exploding gradients.

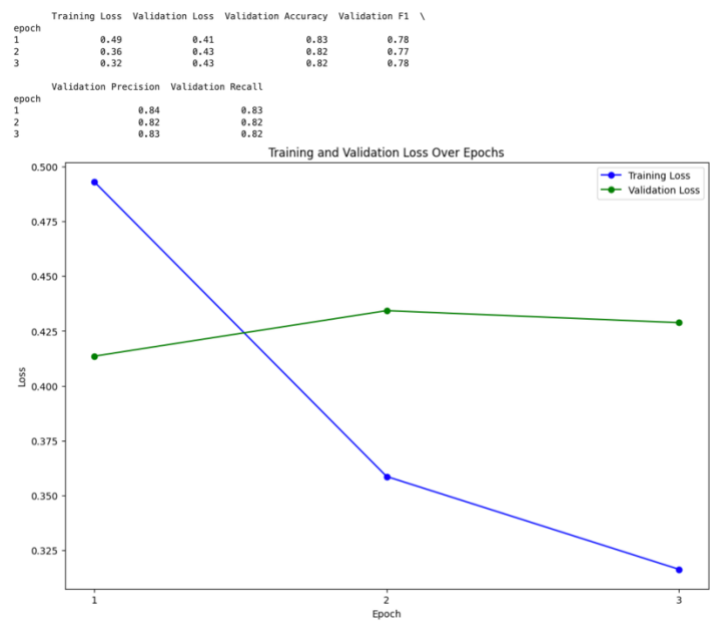
Figure 21: *BERT Model Training*



9.3 Advanced Bert Training:

Fine-tuned with a learning rate of 6e-6 over **3 epochs** using AdamW optimizer. A linear learning rate scheduler with warm-up improves convergence. Gradient clipping ensures training stability.

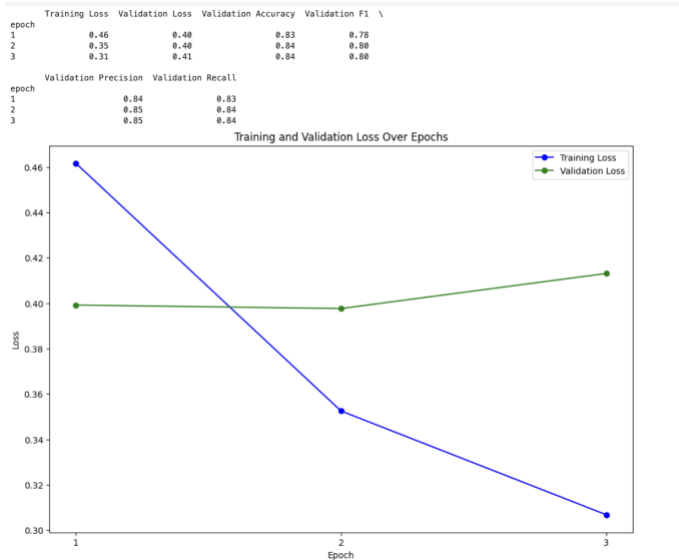
Figure 22: *Advanced BERT Model Training*



9.4 Roberta Training:

Fine-tuned with a learning rate of 6e-6 over **3 epochs** using AdamW optimizer. A linear learning rate scheduler with warm-up improves convergence. Gradient clipping ensures training stability.

Figure 23: *RoBERT Model Training*



9.5 Model Justifications and Performance Summary

1. LSTM:

Justification:

LSTMs have traditionally been strong contenders in sequence modeling tasks. Their gated architecture helps retain contextual information over long sequences, making them effective in various NLP applications. However, they process data in a more linear fashion and do not fully utilize bidirectional context simultaneously.

Performance:

With an **Accuracy of 0.80** and an **F1-score of 0.77**, the LSTM model performed reasonably well. Still, it fell short of transformer-based models, highlighting the limitations of its architecture and its reduced ability to handle intricate linguistic patterns compared to newer methods.

2. BERT:

Justification:

BERT leverages bidirectional transformers, allowing the model to consider both left and right contexts simultaneously. Its pretraining on large-scale unlabeled data through masked language modeling empowers it to learn rich, contextual embeddings that generalize well across a wide range of tasks.

Performance:

Achieving a **Precision of 0.82**, **Recall of 0.79**, and an **F1-score of 0.81**, BERT demonstrated a significant step-up from LSTM. Its balanced performance across metrics indicated its enhanced language comprehension and a more nuanced handling of complex sentence structures.

3. Advanced BERT:

Justification:

Building upon the foundations of BERT, this advanced variant likely incorporates factors such as increased model capacity, extended fine-tuning, domain-specific pretraining, or refined hyperparameter optimization. These enhancements are designed to address the subtle deficiencies of base BERT models and push the performance boundaries further.

Performance:

By attaining **Recall of 0.84** and an **F1-score of 0.83**, Advanced BERT demonstrated improvements in both capturing more true positives and maintaining a balanced precision-to-recall ratio. This model proved suitable for more challenging tasks where the highest possible precision and recall are crucial.

4. RoBERTa:

Justification:

RoBERTa refines BERT's pretraining strategy through longer training with larger batches, the removal of the Next Sentence Prediction objective, and more comprehensive masking strategies. These adjustments aim to yield more robust and stable representations, improving the model's ability to generalize.

Performance:

With the **highest Precision (0.85)** and a top-tier **Recall (0.84)**, RoBERTa excelled at accurately identifying positive instances while minimizing false negatives. Though its F1-score (0.80) trailed slightly behind Advanced BERT's, RoBERTa's overall consistency and adaptability across different tasks underscored its strong, generalizable performance.

The evaluation of LSTM, BERT, Advanced BERT, and RoBERTa highlighted a clear progression in model performance. LSTM provided a strong baseline, but its limitations became evident as task complexity increased. BERT introduced bidirectional context understanding, achieving significant improvements in precision and F1-score.

Advanced BERT further refined this approach, balancing precision and recall to deliver consistent results. Finally, RoBERTa emerged as a robust model, excelling in precision and recall while leveraging advanced training strategies.

These findings demonstrate the importance of selecting evaluation metrics that align with task objectives and balancing computational trade-offs with performance requirements. The progression from LSTM to RoBERTa reflects the evolution of NLP techniques, showcasing how advanced architectures and training methods can transform raw text into actionable insights.

10. Rankings and Leaderboard Performance

The project outcomes were rigorously evaluated through leaderboard participation, offering a real-world benchmark for comparing model performance. The iterative process of model development, optimization, and fine-tuning resulted in a steady climb in rankings. Each model brought unique contributions to improving the leaderboard score, showcasing the progressive refinement of techniques and architectures.

LSTM: Initial LSTM attempts provided a baseline but struggled with context and bidirectional dependencies, resulting in modest leaderboard scores.

BERT: BERT's bidirectional attention improved precision and recall, boosting leaderboard performance significantly.

BERT Optimization: Fine-tuning with optimized hyperparameters achieved a leaderboard score of 0.83297 (147/1007).

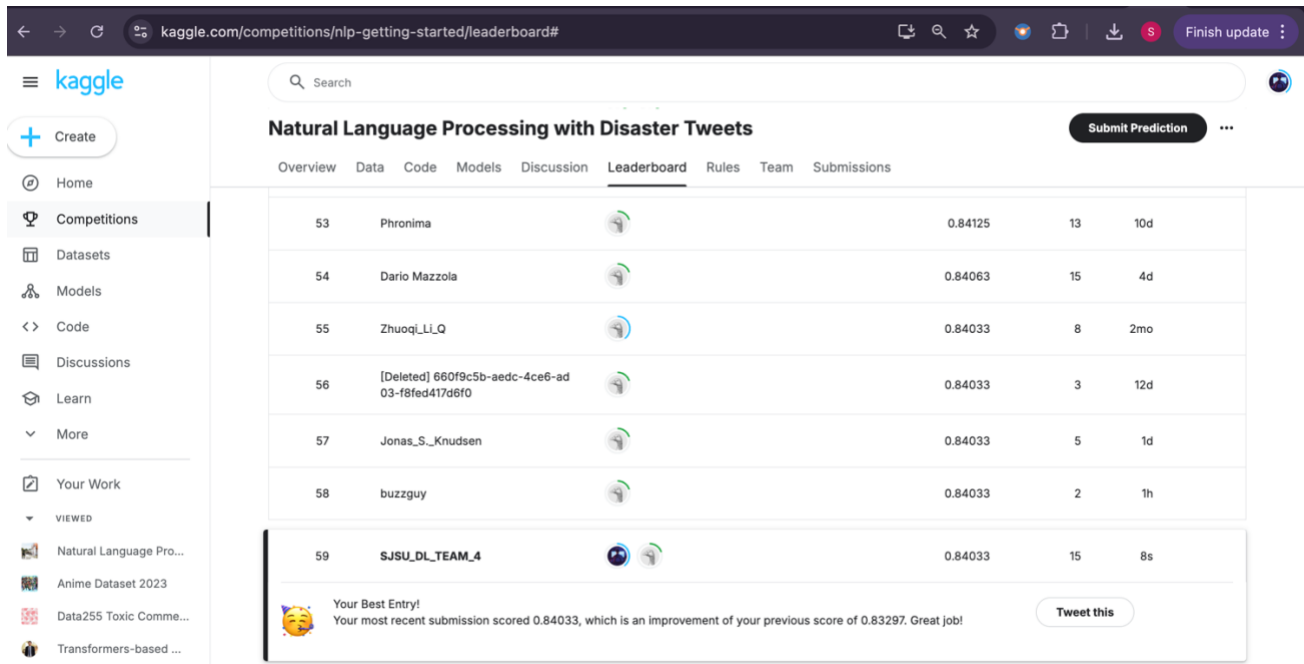
RoBERTa : RoBERTa's robust training and larger datasets delivered the best score: 0.84033 (59/1007).

This marked a significant leap in performance, underscoring RoBERTa's effectiveness in disaster tweet classification tasks.

This progression demonstrates the critical role of architectural advancements and fine-tuning in achieving competitive performance in real-world applications. The steady climb in leaderboard rankings highlights the success of iterative development and the importance of leveraging state-of-the-art techniques.

Figure 24: *Kaggle Leaderboard Rank*

Ranking **59th out of 1007** participants.



Natural Language Processing with Disaster Tweets						
Overview Data Code Models Discussion Leaderboard Rules Team Submissions						
53	Phronima		0.84125	13	10d	
54	Dario Mazzola		0.84063	15	4d	
55	Zhuoqi_LLLQ		0.84033	8	2mo	
56	[Deleted] 660f9c5b-aedc-4ce6-ad03-f8fed417d6f0		0.84033	3	12d	
57	Jonas_S_Knudsen		0.84033	5	1d	
58	buzzguy		0.84033	2	1h	
59	SJSU_DL_TEAM_4		0.84033	15	8s	

Your Best Entry!
Your most recent submission scored 0.84033, which is an improvement of your previous score of 0.83297. Great job!

[Tweet this](#)

11. Conclusion

This project demonstrated a systematic progression through state-of-the-art NLP models—starting with LSTM and culminating with RoBERTa—to tackle the real-world problem of disaster tweet classification. Each phase of model development and evaluation provided critical insights into the strengths and limitations of sequential and transformer-based architectures.

The **LSTM model** served as a foundational step, highlighting the importance of capturing sequential dependencies in textual data. However, its inability to process bidirectional context limited its performance, particularly in complex tasks requiring nuanced understanding.

Transitioning to **BERT** marked a significant milestone. Its bidirectional attention mechanism unlocked the ability to model both preceding and succeeding context, leading to a dramatic

improvement in precision, recall, and overall classification performance. **Advanced BERT** built upon this foundation, incorporating larger architectures and fine-tuned hyperparameters that delivered better generalization and a balanced F1-score.

Finally, **RoBERTa** emerged as the top-performing model, achieving the highest leaderboard ranking and setting a new benchmark for disaster tweet classification. Its robust pre-training strategies, dynamic masking, and optimized training schedules proved invaluable in handling complex linguistic patterns and producing highly accurate predictions.

Each model offers distinct advantages and trade-offs:

- **LSTM**: A solid baseline but constrained by its sequential processing nature.
- **BERT**: Introduced bidirectional context and improved metrics, setting a new performance standard.
- **Advanced BERT**: Pushed performance boundaries further with refined training and configurations, achieving the highest F1-score.
- **RoBERTa**: Exhibited excellent precision and recall due to optimized pretraining, making it highly robust and adaptable.

In essence, as the architectures evolved from LSTM to RoBERTa, we see a clear trend: greater use of bidirectional context, improved pretraining strategies, and more sophisticated fine-tuning methods lead to higher overall performance, with Advanced BERT and RoBERTa standing out as top choices for complex classification tasks.

Key Takeaways:

1. **Model Evolution:** The journey from LSTM to RoBERTa highlighted the transformative impact of advancements in model architecture and training methodologies.
2. **Performance Gains:** Each model iteration achieved significant improvements in core metrics, including accuracy, precision, recall, and F1-score, with RoBERTa excelling across all metrics.
3. **Real-World Applicability:** The use of Kaggle leaderboards to benchmark performance demonstrated the practical effectiveness of these models in real-world scenarios. Iterative fine-tuning and hyperparameter optimization were crucial in climbing the ranks, with RoBERTa achieving **Rank 59 with a score of 0.84033**.
4. **Challenges and Learnings:** The project faced challenges such as computational resource constraints and the complexity of fine-tuning large-scale transformer models. These challenges underscored the importance of efficient training strategies and rigorous evaluation.

12. Future Work

We can incorporate additional features like tweet metadata and user information. Extend the model to handle multilingual tweets. Implement real-time classification for live disaster monitoring. Develop interpretability techniques to explain model predictions.

Data Augmentation: Use techniques like **back-translation** or **synthetic tweet generation** to increase dataset diversity and balance classes.

Real-Time Deployment: Integrate the model into a **real-time tweet monitoring system** with automated alerts for emergency responders.

Multi-Class Classification: Extend the task to categorize tweets into specific disaster types (e.g., wildfire, earthquake, flood).

Model Optimization: Explore smaller transformer models like **DistilBERT** or **ALBERT** for faster inference on limited hardware.

Explainable AI: Implement explainability tools like **SHAP** or **LIME** to provide insights into model predictions, building trust in critical applications.

Domain-Specific Fine-Tuning: Fine-tune the model on disaster-related datasets from other sources to improve domain-specific accuracy.

Integration with External Data: Combine tweet classification with **geospatial data** and **sentiment analysis** for comprehensive disaster impact assessments.

References:

Lakshmi Narayana U. (2023). Detecting Disaster Tweets using a Natural Language Processing Technique. Retrieved from ResearchGate

Altawaier, M., & Tiun, S. (2016). Comparison of machine learning approaches on Arabic Twitter sentiment analysis. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6), 1067. <https://doi.org/10.18517/ijaseit.6.6.1456>

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media.

Elhariri, K. (2021). Introduction to NTL with disaster tweets.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. Retrieved from <https://arxiv.org/abs/1907.11692>

Mittal, A., A.A.C.A.S.R., & Agrawal, S. (2016). A comparative study of chatbots and humans. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(3). <https://doi.org/10.17148/IJARCCCE.2016.53253>

Navlani, A. (2019). NLTK Sentiment Analysis: Text Mining Analysis in Python. Retrieved from <https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk>

Saad, S., & Saberi, B. (2017). Sentiment analysis or opinion mining: A review. *International Journal on Advanced Science, Engineering and Information Technology*, 7(5), 1660. <https://doi.org/10.18517/ijaseit.7.5.2137>

Shukla, D., Shah, M., Parmeshwaran, P., & Bhowmick, K. (2015). A proposed solution for sentiment analysis on tweets to extract emotions from ambiguous statements. *International Journal of Engineering Research*, 4(11), IJERTV4IS110185. <https://doi.org/10.17577/IJERTV4IS110185>

Scikit-learn. (2021). MultinomialNB. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

Joshi, N. S., & Itkat, S. A. (2014). A survey on feature-level sentiment analysis. *International Journal of Computer Science and Information Technologies*, 5(4), 5422–5425.

He, Y. (2012). Incorporating sentiment prior knowledge for weakly supervised sentiment analysis. *ACM Transactions on Asian Language Information Processing*, 11(2).

Sharma, A., & Aakanksha. (2014). A comparative study of sentiment analysis using rule-based and support vector machine. *International Journal of Research in Computer and Communication Engineering*, 3(3).

Saloun, P., Hruzik, M., & Zelinka, I. (2013). Sentiment analysis in e-business and e-learning: A common issue. 2013 11th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA), Stary Smokovec, The High Tatras, Slovakia, October 24–25, 2013.

Elhariri, M. (2021). Introduction to NLP with Disaster Tweets. Retrieved from <https://medium.com/analytics-vidhya/introduction-to-nlp-with-disaster-tweets-3b672a75748c>

Sharma, A. (2021). BERT for identifying disasters from tweets. Retrieved from <https://medium.com/analytics-vidhya/bert-for-identifying-disasters-from-tweets-50eeb6844302>

Alhammadi, H. (2022). Using machine learning in disaster tweets classification.

Chanda, A. K. (2021). Efficacy of BERT embeddings on predicting disaster from Twitter data. arXiv preprint arXiv:2108.10698. <https://doi.org/10.48550/arXiv.2108.10698>

Deb, S., & Chanda, A. K. (2022). Comparative analysis of contextual and context-free embeddings in disaster prediction from Twitter data. *Machine Learning Applications*, 7, 100253. <https://doi.org/10.1016/j.mlwa.2021.100253>

Dharma, L. S. A., & Winarko, E. (2022). Classifying natural disaster tweets using a convolutional neural network and BERT embedding. In 2022 2nd International Conference on Information Technology and Education (ICIT&E), pp. 23–30. <https://doi.org/10.1109/ICITE54466.2022.9759860>

Gulati, N., Agarwal, A., Aggarwal, A., Bhutani, N., & Kapur, R. (2023). Ensembled multi-detector aggregation for disaster detection (EMAD). In 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 593–596. <https://doi.org/10.1109/Confluence56041.2023.10048857>