Disaster Detection: Harnessing NLP and Machine Learning for Crisis Classification in Social Media

Arham Hundia
hundia@purdue.edu

Parth Patel
patel384@purdue.edu

Sowmya Jayaram Iyer
jayarami@purdue.edu

Abstract

In an era of rapid information dissemination, social media platforms such as Twitter have become invaluable resources for understanding public sentiment and identifying crucial real-world events. As the volume of tweets surpasses 6000 per second, the potential for timely detection of emergency situations and natural disasters grows ever more significant. Leveraging advances in Natural Language Processing (NLP) techniques, our research aims to enhance disaster response by accurately and swiftly classifying tweets related to genuine emergencies or calamities. We employed a dataset comprising 10,000 manually classified tweets, created as a Kaggle dataset, to construct and evaluate various models, TFdif vectorisation with SVM, Glove embeddings with DAN, RNN, LSTM and BiLSTM and BERT tokenisation with BERT. Our findings indicate that the BERT delivers the most promising performance, achieving an f1 score of 0.83, thereby demonstrating the potential of NLP-driven disaster tweet classification to save lives through expedited response times. GitHub (https://github.com/parthpatel0329/Disaster-Tweet-Classification) Slides

1. Introduction

In the era of social media, the efficient filtering and utilization of vast amounts of data for disaster prediction and response is crucial. Our project **focuses** on a specific problem: accurately determining whether a tweet pertains to a genuine disaster or not, using a dataset of 10,000 hand-classified tweets from a Kaggle competition. This ***problem is distinct*** from other related works because it emphasizes the importance of word sense disambiguation in the context of disaster management.

To address this problem, we explored various machine learning and natural language processing techniques, such as SVM, DAN, RNN, LSTM, BiLSTM, and BERT. **Alternative techniques** for disaster tweet classification, such as rule-based systems, Naive Bayes, k-NN, and Random Forest, may not provide comparable performance or scalability to deep learning models like LSTM and BERT. Deep learning techniques excel in handling sequential data, leveraging pre-trained embeddings, addressing language ambiguity, and adapting to evolving language patterns, which are essential for accurately classifying disaster-related tweets. Our **research demonstrates** their state-of-the-art performance and what makes them more suitable for the specific task of disaster tweet classification compared to the alternatives mentioned.

Our approach also stands out because we employed data cleaning techniques, including tokenization, stop-word removal, stemming, and lemmatization, and visualizations like term frequencies and keyword counts for deeper insights.

The **contributions** of this paper aim to answer the following questions:

1. How can disaster-related tweets be accurately classified using various machine learning and natural language processing techniques?
2. How do different techniques perform in the classification task, and which method is the most effective?
3. How do simpler models like SVM with TF-IDF perform compared to more complex models like BERT and LSTM?
4. What are the limitations of the explored models, and how can they be improved for better performance?

The **risks involved** in our research are the limitations of the explored models, which can impact their performance in disaster tweet classification. On the other hand, the **opportunities include** improvements in these models, leading to more accurate classification results and, ultimately, more effective disaster management and response strategies.

The analysis of social media data, particularly tweets, for disaster management has been a subject of extensive research in recent years. This section highlights key studies that have contributed to the field and shares similarities with our work. In 2013, [3] investigated the potential of machine learning classifiers in extracting informative tweets during crises. They compared the performance of several algorithms, including Naive Bayes, Decision Trees, and SVM, and discussed the challenges of processing and analyzing social media data in emergency situations.

[6] examined the utility of deep learning models, specifically Convolutional Neural Networks (CNN), for disaster-related tweet classification. Their work demonstrated the superior performance of CNNs over traditional machine learning algorithms in identifying informative tweets during disasters. [1] introduced BERT in 2018, a pre-trained language model that has significantly improved NLP tasks, including text classification. Their study has inspired the adoption of BERT in various contexts, including disaster tweet classification, as evidenced by our research. [5] assessed the performance of BERT in the context of disaster tweet classification and introduced CrisisBERT, a BERT-based model fine-tuned on crisis-related tweets. Their study demonstrated the effectiveness of transfer learning and BERT in identifying disaster-related information on social media.

Our **research builds upon** the previous works by comparing various models and techniques, TF-IDF features with SVM and LSTM models, GloVe vectorization with DAN, RNN, LSTM and BiLSTM, and BERT tokenisation with BERT. We **contribute** to the body of knowledge by demonstrating the superior performance of the BERT model in the context of disaster tweet classification, which has implications for more efficient disaster response strategies. Our study reveals that the BERT model, outperforming the LSTM with pre-trained GloVe Twitter embedding, yields the best results, achieving an average F-1 score improvement of 0.17.

2. Methods

2.1. Preprocessing

Before delving into the classification of disaster-related tweets, we preprocessed the data to **ensure consistency and reduce noise**. This involved converting the text to lowercase, removing non-alphabetic characters, performing lemmatization, and tokenizing the text. Additionally, we experimented with various tokenizers and word embeddings, such as count vectors, TF-IDF vectorization, and GloVe. We also employed pretrained BERT tokenizer for BERT training. This critical preprocessing stage prepared the data for further analysis and modeling.

2.2. TF-dif and SVM

The TfidfVectorizer is a text feature extraction technique that transforms raw text data into numerical feature vectors. It combines the term frequency with the inverse document frequency (how common the word is across all documents in the dataset). This **helps weigh down the importance of common words** and increase the importance of rare but significant words, resulting in more informative features for text classification tasks. Support Vector Machines (SVM), in the context of disaster tweet classification, work by finding an optimal hyperplane that best separates the data points belonging to different classes, in this case, disaster-related and non-disaster-related tweets. By maximizing the margin between the classes, SVM minimizes classification errors while providing a robust and effective model. SVM is unable to capture contextual information. SVMs can struggle with high-dimensional, sparse text data, resulting in suboptimal classification performance. The SVM was trained using the sklearn library.

2.3. Glove Embeddings

GloVe (Global Vectors for Word Representation) is a widely-used unsupervised learning technique for generating word embeddings. Developed by Stanford researchers [7], GloVe embeddings **capture semantic and syntactic information** of words by leveraging the co-occurrence statistics in large corpora. The primary idea behind GloVe is that the ratio of co-occurrence probabilities of word pairs encodes a significant amount of **semantic meaning** which is essential for context based word sense disambiguation.

2.4. DAN

DAN (Deep Attention Neural Network) [4] **provides a baseline as to how word embeddings will perform on a feed-forward fully-connected neural network**. It operates by first converting input text into word embeddings, and then computes the average of these word embeddings. This averaged vector is subsequently passed through a series of fully connected layers, also known as dense layers, with non-linear activation functions such as ReLU. Finally, the output layer, often using a softmax activation function, produces probabilities for each class, allowing the model to make predictions. In the context of our paper the DAN architecture is made up 3 layers. The input layer is the average of the glove embeddings, the hidden layer is made up of 50 neurons, and the output layer has 1 neuron. The problem pertaining to DAN is that it is unable to capture contextual information. DAN was trained on our own using the torch library.

2.5. RNN

RNN (Recurrent neural network) [9] are designed to process sequential data, with connections that allow information to persist from one step of the sequence to the next. This is crucial, as it allows them to **capture the context and dependencies** within the text, leading to a better understanding of the disaster-related content and improved classification accuracy. RNN consists of an input layer, a hidden layer with recurrent connections and an output layer. RNNs use BPTT (backpropagation through time), during training, to minimize a loss function and update the weights. One problem pertaining to RNNs is the fact that they can suffer from **vanishing gradient problems** and can have **difficulty in learning long-range dependencies** . In the context of our paper the RNN architecture is made up 3 layers. The input layer is the RNN layer which takes glove embeddings as inputs, the fully connected layer is made up of 50 neurons, and the output layer has 1 neuron. RNN was trained on our own using the torch library. Computational implementation issues with DANs may arise from the need to process high-dimensional, dense input representations, which can increase memory requirements and computational complexity.

2.6. LSTM

LSTM (Long Short-Term Memory) [2] networks are a type of RNN architecture that use gated cells to regulate the flow of information and overcome the vanishing gradient problem in standard RNNs. LSTMs effectively handle **long-range dependencies** in text, enabling accurate disaster tweet classification by **capturing crucial contextual information across varying sequence lengths**. It consists of three gates (input gate, forget gate, and output gate) and a memory cell. Memory cell is where the gates determine how much new information should be added to or removed from the cell. One variant of LSTM architecture is Bi-LSTM (Bidirectional LSTM) that processes the input sequence in both forward and backward directions using two LSTMs, and concatenates their output to incorporate information from both past and future contexts. BiLSTMs **improve classification accuracy by considering both past and future** words within the sequence. **BiLSTMs can be computationally intensive**, as they require simultaneous processing of both forward and backward directions, potentially increasing training time. In the context of our paper the LSTM architecture is made up of 3 layers. The input layer is the LSTM layer which takes glove embeddings as inputs, the fully connected layer is made up of 50 neurons, and the output layer has 1 neuron. In the context of our paper, the Bi-LSTM architecture is made up of 3 layers. The input layer is the Bi-LSTM layer which takes glove embeddings as inputs, the fully connected layer is made up of 50 neurons, and the output layer has 1 neuron. The LSTM and BiLSTM were trained on our own using the torch library.

2.7. BERT

BERT, or Bidirectional Encoder Representations from Transformers, [8] is based on the Transformer architecture pre-trained on large text corpora using unsupervised tasks like Masked Language Modeling to learn language structure and semantics. **BERT demands significant computational resources and memory for training**. Hence we use pre-trained BERT. BERT leverages **pre-trained embeddings and a deep bidirectional transformer architecture**, allowing it to **better capture complex semantic relationships in disaster tweets**, resulting in superior classification performance. In the context of this paper, we used the pre-trained BERT-base-uncased model, which is a smaller version of BERT consisting of 12 layers (transformer blocks), 12 attention heads, and 110 million parameters. The model utilizes the ”uncased” tokenization approach, meaning that all text is converted to lowercase before being processed. This makes the model more robust and less sensitive to case variations in the input text. Transfer learning was used to train BERT where the encoder and pre-trained weights were imported from the transformers library and a custom training pipeline was built in python.

3. Experiments

We use the TfidfVectorizer to transform the raw text data into numerical feature vectors, which will be used as input for the LinearSVC (Support Vector Classifier with a linear kernel) model. We obtain a test accuracy of 0.79 Figure 1. The classification report reveals a precision, recall, and f1-score of 0.81, 0.88, and 0.84 for the non-disaster category, and 0.81, 0.71, and 0.76 for the disaster category. These results indicate that the classifier performs well. When evaluating the model on the training data, the classification report shows an impressive accuracy of 0.976 Figure 1. The reduced performance on the test data could be indicative of *overfitting*.

For the **DAN model**, we used the Stochastic Gradient Descent optimizer with a learning rate of 0.1, and a batch size of 30, and trained the model for 150 epochs. We also utilized binary cross entropy as the loss function. The final classification report shows an overall accuracy of 0.59 on test data Figure 1 with an F1-score of 0.421. The results obtained from our experiments demonstrate the ineffectiveness of the DAN model in tasks that require context.

For the RNN model, we used the Stochastic Gradient Descent optimizer with a learning rate of 0.1, a batch size of 30, and trained the model for 25 epochs. We also utilized binary cross entropy as the loss function. The final classification report shows an overall accuracy of 0.745 on test data Figure 1 with an F1-score of 0.674. The results obtained from our experiments demonstrate the effectiveness of the RNN model in language processing tasks.

For the LSTM model, we used the Stochastic Gradient Descent optimizer with a learning rate of 0.1, a batch size of 30, and trained the model for 100 epochs. We also utilized binary cross entropy as the loss function. The final classification report shows an overall accuracy of 0.729 on

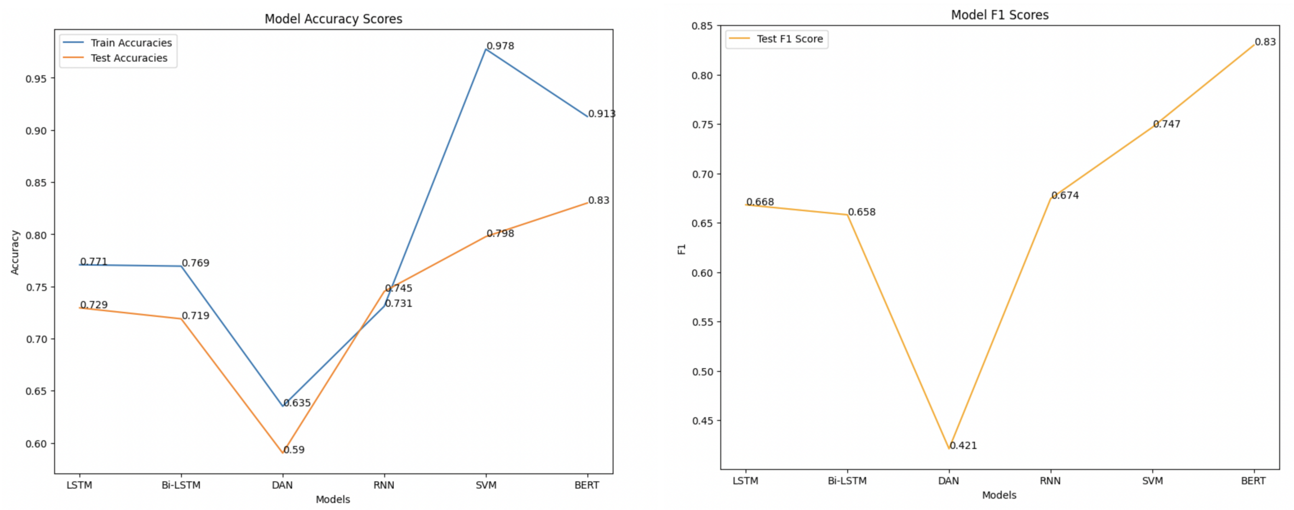

Figure 1. (Left) Comparison of Train and Validation Accuracies (Right) Comparison of Validation F1 scores

test data Figure 1 with an F1-score of 0.668. The results obtained from our experiments demonstrate the effectiveness of the LSTM model in language processing tasks along with understanding context.

For the Bi-LSTM model, we used the Stochastic Gradient Descent optimizer with a learning rate of 1, a batch size of 30, and trained the model for 20 epochs. We also utilized binary cross entropy as the loss function. The final classification report shows an overall accuracy of 0.719 on test data Figure 1 with an F1-score of 0.658. The results obtained from our experiments demonstrate the effectiveness of the Bi-LSTM model in language processing tasks along with understanding context in both directions for the given input.

For the pre-trained BERT model, we used the AdamW optimizer with a learning rate of 2e-5, a batch size of 16, and trained the model for 3 epochs. We also utilized a linear learning rate scheduler with a warmup period of 0 steps. The final classification report shows an overall accuracy and F-1 score of 0.83 on test data Figure 1 with an F1-score of 0.85 for class 0 (non-disaster) and 0.80 for class 1 (real disaster). The results obtained from our experiments demonstrate the effectiveness of the BERT model with attention mechanism in the binary disaster classification task.

3.1. Results and key-takeaways

After implementing initial models, a comprehensive error analysis revealed key conclusions. The limited training dataset size (6,000 records) negatively impacted model accuracy, causing overfitting. Ambiguity, subjectivity, and the need for additional context affected performance. For example, a tweet discussing a music video with burning buildings and police chases was incorrectly labeled as a disaster-related tweet due to the presence of disaster-related terms. Non-ASCII characters and emojis' absence led to misinterpretations, and semantic misconstruals like sarcasm and metaphors complicated classification. For instance, a tweet "HALSEY AND TROYE COLLAB WOULD BE BOMB" was prone to be incorrectly classified as disaster-related. Misleading keywords and hashtags caused misclassification, and the short length of tweets posed challenges due to limited context and information.

In conclusion, our experiments with various models on the binary disaster classification task demonstrate the varying levels of effectiveness of each approach 1. The LinearSVM model shows good performance, with an overall test accuracy of 0.79, while the DAN model falls short with a test accuracy of 0.59. The RNN and LSTM models exhibit better language processing capabilities with test accuracies of 0.745 and 0.729, respectively. The Bi-LSTM model further enhances the understanding of context by taking into account bidirectional information, achieving a test accuracy of 0.719. Lastly, the pre-trained BERT model outperforms all other models, achieving an impressive test accuracy of 83%.

The above results indicate that DAN is not context-aware making its performance significantly lower than that of other models. The Linear SVM performance was very unexpected. The huge disparity in the train and test accuracy in Figure 1 shows that the Linear SVM was overfitting extensively on the training dataset, but was able to generalize better on the test dataset. The RNN model on the other hand was performing on par with the LSTM and Bi-LSTM due to the LSTM and Bi-LSTM models overfitting on the training dataset. This also explains why Bi-LSTM was performing about the same as the LSTM. This can be attributed to the small size of the training dataset.

The BERT model's performance can be attributed to its attention mechanism. Notably, BERT is pre-trained on a vast amount data while all the other models used only pre-trained word embeddings. This transfer learning equips BERT for fine-tuning tasks like disaster detection. Overall, our results demonstrate the effectiveness of BERT in NLP tasks that require a deeper understanding of context and reaffirm its position as state-of-the-art. For the sake of time, we decided to not implement Logistic Regression, mentioned in the proposal, because we believed it was too naive of an approach for this problem.

4. Discussion

In our study, we addressed the following questions: (1) We demonstrated that disaster-related tweets can be accurately classified using machine learning and natural language processing techniques, such as SVM, RNN, LSTM, Bi-LSTM, and BERT. (2) Our results showed that the BERT model outperformed all other models, making it the most effective method for this task. (3) While the simpler SVM model with TF-IDF achieved good performance, it was outperformed by more complex models like BERT, which were better at handling context and semantics. (4) Limitations of the explored models included context insensitivity (DAN), overfitting (SVM), and limited dataset size. These issues can be addressed by refining the dataset, incorporating advanced pre-processing techniques, and using more advanced NLP models that capture richer contextual information.

Through our experiments with various models for binary disaster tweet classification, we gained valuable insights into the challenges and complexities of the problem. The main drivers that contributed to the effectiveness of certain models over others can be attributed to their ability to handle context, capture intricate language structures, and understand semantics in text data.

We identified several factors that impacted the performance of our models, such as ambiguity, subjectivity, limited dataset size, non-ASCII characters and emojis, and factors like sarcasm. The short length of tweets themselves also posed challenges for accurate classification due to their limited context and information.

If given more time to work on this project, we would explore additional approaches and improvements:

1. **One month:** Refine and expand the dataset by collecting more tweets and incorporating external context, such as images or links, to enhance the model's understanding. Investigate the use of advanced preprocessing techniques and ensemble methods to improve model performance.

2. **One year:** Leverage more advanced NLP techniques, such as attention mechanisms and context-aware embeddings (e.g., ELMo) to capture richer contextual information. Develop a evaluation framework to ensure robustness and applicability in real-world scenarios.

3. **Five years:** Address inherent challenges of language, such as ambiguity and semantics, and integrate these insights into our models. Investigate unsupervised or semi-supervised learning methods for large-scale disaster tweet classification. Collaborate with experts in disaster management to develop real-time disaster response strategies.

5. Acknowledgement

Our group consists of Arham Hundia, Parth Patel and Sowmya Jayaram Iyer. Arham Hundia played a crucial role in the data preprocessing phase and the implementation of the SVM model. Parth Patel was responsible for the development of RNN and LSTM models, while Sowmya Jayaram Iyer focused on BERT and LSTM models.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[2] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 2012. 3

[3] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. Aidr: Artificial intelligence for disaster response. Association for Computing Machinery, 2014. 2

[4] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015. 2

[5] Junhua Liu, Trisha Singhal, Lucienne TM Blessing, Kristin L Wood, and Kwan Hui Lim. Crisisbert: a robust transformer for crisis classification and contextual crisis embedding. In *Proceedings of the 32nd ACM conference on hypertext and social media*, 2021. 2

[6] Dat Nguyen, Kamela Ali Al Mannai, Shafiq Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. Robust classification of crisis-related data on social networks using convolutional neural networks. In *Proceedings of the international AAAI conference on web and social media*, 2017. 2

[7] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014. 2

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[9] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. 1989. 2