Enhanced Verification of Natural Disaster Tweets: A Dual-Layered Approach for Improved Accuracy and Reliability

Hanin Atwany, George Ibrahim, Ching Chao
Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
{Hanin.Atwany, George.Ibrahim, Ching.Chao}@mbzuai.ac.ae
G-29

Abstract

In the era of digital communication, Twitter stands out as a vital tool for spreading real-time information during natural disasters, however, this can be a way to spread fake news. This project introduces a two-layered approach to firstly identify if a tweet relates to a natural disaster and secondly to verify its accuracy by comparing it with web-based sources, a process known as stance classification. Stance classification involves assessing whether the content of a document supports or refutes a specific claim. Our first Twitter-RoBERTa model, which aims to classify a tweet as a natural disaster, demonstrated an 82.81% accuracy on the test set and an 82.653% Kaggle public score, surpassing the baseline score by approximately 4.9%. For the stance classification layer, we utilized an additional RoBERTa model. This model played a crucial role in instances where the first layer had inaccurately classified a tweet. In such cases, the second layer effectively rectified these misclassifications, achieving an accuracy rate of 76.0%. The Code is available at https://github.com/GeorgeSherif/Enhanced-Verification-of-Natural-Disaster-Tweets.

1 Introduction

Social media significantly influences our perspectives and behaviors in the modern world. People frequently accept information in tweets as factual without verifying its accuracy. Rumors are diverse, ranging from entirely false to subtly altered and misleading. Specifically, in the context of natural disasters, complex information is prone to manipulation by those who spread rumors.

The abundance and timeliness of social media posts, particularly during emergencies, can significantly aid in decision-making processes [13]. These posts often provide immediate, practical information that is crucial for effective emergency response. Furthermore, real-time reporting of emergencies, akin to the way incidents are shared on platforms like Twitter, enhances emergency response and disaster management strategies [14, 21].

Social media, while crucial and impactful in natural disaster scenarios, can also be utilized in negative ways. During the US presidential elections involving Donald Trump and Hillary Clinton, social media was widely used to share false news about both candidates, reaching millions of shares [4]. Similarly, in the 2021 US presidential campaign, newer studies revealed even larger misinformation campaigns related to COVID-19.

Silverman et al. [19] curated datasets of highly shared true and false political stories on Twitter, providing valuable resources for evaluating fake news detection methods. Pivotal work by Vosoughi et al. explored the spread of false information on Twitter, emphasizing the need for effective detection techniques [23]. Additionally, research on the credibility of Twitter stories by Castillo et. al

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

[9], and the analysis of rumor propagation on Twitter by Bodaghi et al. [1] have contributed to our understanding of the challenges and opportunities in this domain.

Recent studies show that Twitter is a key platform for real-time information during disasters, but it also faces issues with fake news and misinformation. Our project addresses this by developing a two-step tool that first determines if a tweet concerns a natural disaster and then verifies it against recent Google Search documents, effectively conducting stance classification.

Contributions

- Leveraged BERTopic to capture hidden topics in the text, providing additional contextual information and enhancing the model's understanding of subtle relationships
- Introduced a second layer of stance classification to double-check and verify the first layer's output, offering the proof link for validation.

2 Related Work

2.1 Layer 1: Natural Disaster Detection

Research in classifying disaster-related tweets, including those about COVID-19, has advanced with techniques like machine learning and NLP. Muller et al. introduced COVID-Twitter-BERT (CT-BERT), a model trained on 160 million COVID-19-related tweets, designed for classification, question-answering, and chatbots [18]. CT-BERT outperformed BERT-LARGE by 10-30% in COVID-19 content, underscoring the value of domain-specific models in NLP, especially for social media analysis.

Barbouch et al. developed a method to categorize tweets about natural disasters, considering both tweet content and user influence [6]. They compared traditional text classifiers with BERT, a transfer learning model, and incorporated the users' Twitter network rank in classification. Results indicated BERT's superiority over traditional methods, particularly in larger categories, and highlighted the significance of user influence based on social standing in certain tweet categories.

2.2 Layer 2: Detecting Fake News through Stance Classification

Detecting fake news presents a challenge in the field of Natural Language Processing. Ferreira et al. have created "Emergent", a new dataset, labeled by journalists, for rumor refuting, which includes 300 rumored claims and 2,595 related news articles [11]. They employed a logistic regression classifier to determine the stance of an article's headline relative to the claim. Their method attained an accuracy of 73%, which is significantly higher (by 26%) than the accuracy achieved by the Excitement Open Platform [17].

Hanselowsk et al. reproduced the experimental setup for the 2017 Fake News Challenge Stage 1 (FNC-1) task, which tackled the stance classification task as a necessary step towards detecting fake news [12]. The authors identify a major flaw in FNC-1's evaluation metric, showing it favored majority classes and suggesting a new F1-based metric for more accurate system ranking. They also introduced a feature-rich stacked LSTM model which was superior in predicting minority classes.

Toward detecting fake news on Twitter, Buntain et al. trained two models on the CREDBANK and PHEME datasets for crowdsourced and journalistic accuracy assessments, respectively [8]. They then tested these models on a third dataset, BuzzFeed's fake news dataset. The authors concluded that the model trained on the crowdsourced dataset (CREDBANK) was more efficient than the one trained on the journalistic assessments (PHEME) in detecting fake news on Twitter.

3 Problem Statement

The rapid dissemination of information through social media, especially during natural disasters, presents a unique challenge: the need for quick verification of the authenticity and accuracy of the information being shared. This project addresses the critical issue of identifying and verifying tweets related to natural disasters.

The significance of this problem lies in the potential impact of misinformation during natural disasters. False reports can lead to unnecessary panic, misallocation of resources, and even harm to those affected by the disaster. Conversely, timely and accurate information can aid in effective disaster response and management.

Our project aims to develop a system that detects tweets pertaining to natural disasters and then cross-verifies these tweets using stance classification techniques applied to articles scraped from the web. This approach intends to assess the alignment of information in tweets with that in reputable news sources, thereby evaluating their credibility. By automating this process, the system seeks to provide a rapid and reliable way of filtering and verifying information during critical times of natural disasters.

4 Proposed Method

The flowchart in Figure 1 shows the architecture of our model, and how the 2 layers interact with the tweet, label, and the Google API.

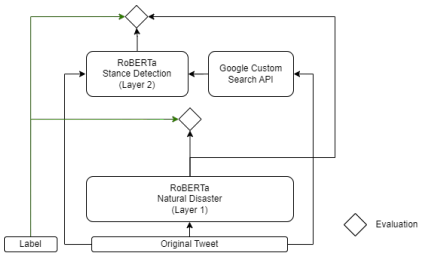

Figure 1: Model Architecture

4.1 Layer 1: Natural Disaster Detection

Integrating BERTopic-Enhanced Sentiment Analysis Architecture

Building upon the pre-trained Twitter-RoBERTa model, our architecture has been enhanced to incorporate valuable topic information extracted using BERTopic. This augmentation is achieved by introducing an additional feature (topics) into the classification model. The process unfolds in two key steps:

In the $CaseDataset$ class, we seamlessly integrate BERTopic-derived topics into the dataset creation process. Within the $getItem$ method, the topics are extracted from the DataFrame and included as part of the input dictionary. This dictionary, comprising review text, input IDs, attention masks, targets, and topics, serves as the foundation for training and validation data. The integration of cross-entropy loss further refines the model's training process, guiding it to minimize the dissimilarity between predicted and true sentiment labels.

Within the $SentimentClassifier$ class, topics are effectively incorporated into the model's architecture. In the forward method, the extracted topics are concatenated along the last dimension with the output from the RoBERTa model. This concatenated tensor undergoes further processing through linear and activation layers, culminating in the production of the final classification output. The use of cross-entropy loss 1 as the optimization criterion ensures that the model learns to predict labels with a focus on minimizing the dissimilarity between predicted and true probability distributions.

$$CrossEntropyLoss(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \cdot \log(\hat{y}_{ij}), \tag{1}$$

In essence, our refined model architecture synergizes the inherent contextual understanding of the Twitter-RoBERTa model with the enriched feature set derived from BERTopic-assigned topics, leveraging cross-entropy loss to guide the training process effectively.

4.2 Layer 2: Detecting Fake News through Stance Classification

Our second layer consists of another RoBERTa model, which has been trained using a Stance Classification dataset, known as FEVER [20]. We used the output of the first layer, to retrieve the most relevant document for each tweet. Now, the modified dataset can go through the second layer for stance classification.

This stage is crucial for verifying the tweet's content against retrieved documents gathered from the internet. This process allows us to determine the tweet's accuracy and consistency with reported news, which serves as an initial step in the detection of fake news.

RoBERTa Using Similar Dataset

Since our dataset was not initially compiled for fact-checking purposes, we fine-tuned RoBERTa using the FEVER dataset. Recent studies have indicated that RoBERTa and BERTweet are the top performers in handling social media datasets [10]. However, given that FEVER is a more general dataset, we opted to use RoBERTa in our approach.

Document Retrieval

To be able to retrieve the most relevant document for each tweet, we utilized Google's Custom Search JSON API. Each tweet is now used as a claim to be validated with the retrieved document. We focus on the top result from Google Search, as it is typically the most relevant to the query.

5 Experimental Setup

5.1 Layer 1: Natural Disaster Detection

During the fine-tuning process, we opted for the T4x2 GPU available on Kaggle. We also used several Python packages and libraries, including pandas, re, nltk, BERTopic, transformers, torch, numpy, sklearn, wandb (Weights & Biases), and Matplotlib. For pre-trained transformer-based language models, we chose the Cardiff NLP model, which is available via the Hugging Face transformers library. The specific model we used was "cardiffnlp/twitter-roberta-base-sentiment-latest."

The baseline model employed an architecture based on RoBERTa. The model consisted of a RoBERTa base with a classification head, utilizing a tanh activation function. It used the AdamW optimizer with a learning rate of 3e-5 and was trained for four epochs. The model achieved a test F1 score of 0.7943. In our enhanced approach, we fine-tuned the RoBERTa model, incorporated data augmentation techniques, and introduced BERTopic for feature extraction, improving the accuracy.

Hyperparameter Tuning

To optimize our classification model, we executed a grid search using Stratified K-Fold cross-validation. This method ensured balanced class representation during evaluation. The search encompassed 24 experiments, key adjustments included setting the learning rate to 3e-6, the batch size to 32, and limiting the maximum sequence length to 128 words. These parameter choices strike an optimal balance between model expressiveness and training efficiency. The grid search, leveraging five folds in Stratified K-Fold, was completed in approximately 9 hours. We then fine-tuned our model by tweaking other factors as well, like dropout rates, trying values 0.1, 0.2, and 0.3. The best accuracy was observed at a dropout of 0.1.

Data Augmentation

Given the inherent variability and informal language usage in tweets, data augmentation becomes crucial for capturing the diverse linguistic patterns present in the data. By augmenting the training dataset through methods like deletion and swapping, we introduce variations in the text, simulating different ways users may express similar sentiments or convey the same information. Deletion involves removing certain words or phrases from the text, and emulating scenarios where users provide concise or incomplete information. Swapping, on the other hand, entails rearranging words or phrases, reflecting the fluidity and flexibility in language expression commonly observed in tweets. We chose those two methods specifically because for a small dataset, these two sub-methods gain

higher improvements than the other two sub-methods that are based on synonym replacement and insertion [15, 24].

Feature Generation

The BERTopic model was employed to extract latent topics from the preprocessed text data. Due to the limited length of tweets, extracting meaningful information from them is challenging. This is because short texts often lack co-occurrence patterns and context information, which are crucial for understanding the underlying meaning [3]. While the overall classification task falls under supervised learning, this topic identification step utilizes unsupervised learning techniques. The extracted topics were then added as an extra feature. These enhancements aimed to enrich the model's understanding by incorporating latent topics into the classification process, capturing nuanced relationships between topics and target labels. This combined approach harnesses the strengths of both topic modeling and state-of-the-art language models, offering a more comprehensive representation of the underlying structure in the text data. This approach achieved the highest training accuracy of 87 % on validation data after augmentation as shown in Figure 3.

Additional Experiments

We investigated the impact of different optimizers on the model's performance. Comparing AdamW to SGD, we observed a substantial difference in accuracy.AdamW has a faster running time, low memory requirements, and requires less tuning than any other optimization algorithm. Using SGD resulted in a lower accuracy of 82.53%, contrasting with the higher accuracy of 84.83% achieved with AdamW on the validation dataset as shown in Figure 2. This was expected because of the adaptive learning rates and regularization in AdamW which could help navigate the challenges posed by such data [16].

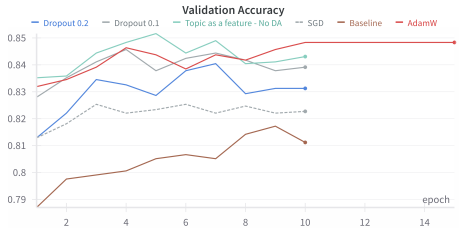

Figure 2: Validation Accuracy for various Dropouts and optimizers.

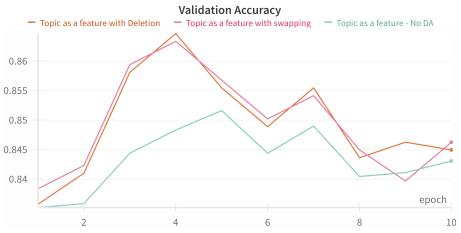

Figure 3: Validation Accuracy with topic as a feature and data augmentation methods.

5.2 Layer 2: Detecting Fake News through Stance Classification

Experiments for this layer were run on HPCs utilizing the A100 GPU. Python packages and libraries were used in this layer including pandas, re, nltk, transformers, torch, numpy, sklearn, and datasets library from Hugging Face. The pre-trained model used for fine-tuning was *roberta-base*. The model was trained for 16 epochs, and the batch size was 16. The learning rate was fixed at $2e-5$. We adjusted the *max_length* parameter to 350, rather than the standard 512, as tweets are typically brief, and answers are often found in the initial sentences of the retrieved document. This decision was based on multiple experiments with various *max_length* values, where 350 yielded the best scores. Fine-tuning the model took slightly less than 2 hours.

5.3 Datasets

We used two datasets, one for each layer. For the first layer, we utilized a dataset sourced from the Kaggle contest [2]. It comprises 7613 entries in total, distributed across five columns. The dataset was divided into training, validation, and test sets, following an 80:10:10 ratio. The dataset includes 4 features, including the ID, Keyword, Location, and the Tweet itself.

Our prediction is whether a given tweet is about a real disaster (1) or not (0). The keyword column has 222 unique values, whereas the text column has 7503 unique tweets. For the location feature, 33% of the rows are nulls. The remaining 66% of the rows have other location values.

For the second layer, we used the FEVER dataset [20]. The dataset was used directly from the $datasets$ library, which is a part of the Hugging Face ecosystem. The $fever - evidence - related$ dataset originally included 485K entries, but for experimental purposes, we fine-tuned our model on a smaller subset consisting of 10,000 entries. The split followed the 80:10:10 ratio in the train, validation, and test splits. This dataset only has the claim and evidence as features, and the output represents whether this claim is supported by the evidence (1) or not (0). Around 73% of the output was 0, which means that the claim was not supported. In addressing dataset imbalance, we conducted experiments using both balanced and unbalanced datasets. While it's a standard practice to balance datasets, certain studies have indicated that this approach may increase accuracy, but often at the cost of reduced recall and precision [5]. Some text pre-processing was performed on the tweets, such as removing the links, hashtags, and other special characters.

5.4 Evaulation Metrics

In our classification-focused project, we evaluated our model using accuracy, precision, recall, and F1-score. Accuracy indicated overall classification correctness. Precision and recall highlighted the model's success in correctly identifying relevant cases and avoiding false negatives. The F1-score provided a balanced measure of precision and recall, useful for imbalanced datasets. These combined metrics offered a thorough assessment of our model's classification ability.

6 Results and Discussion

6.1 Layer 1: Natural Disaster Detection

Accuracy	Precision	Recall	F1-Score
0.8281	0.8161	0.7737	0.7943

Table 1: Results for Layer 1

Table 1 shows the results of the evaluation metrics on the test data from the original data split. Since it was a Kaggle contest, we also had the public score given on hidden test data. We have done several submissions, scoring 0.82653 as our best score. This is around 4.9% higher than the baseline model, which only achieved a 0.77719 score in the leaderboard [22]. This improvement was driven mainly by leveraging an approach that combines BerTopic for feature extraction and RoBERTa for classification. Adding BERTopic to our model improved accuracy by introducing hidden topics present in the text. These topics capture detailed relationships and context, enhancing the model's understanding. The extra information helps the model identify subtle patterns, reducing overfitting and improving predictions on new data.

We explored deletion and swapping as augmentation methods and found that both approaches resulted in very similar and enhanced model performance. The similarity in performance between the two methods indicates the model's robustness in handling variations introduced by both augmentation techniques.This could be attributed to the fact that random deletion introduces natural variations in text, reflecting real-world scenarios of incomplete or corrupted data which is common in tweets. [7]. Figure 3 illustrates a performance comparison between the model without augmentation and the same model with the incorporation of two distinct augmentation methods.

6.2 Layer 2: Detecting Fake News through Stance Classification

The evaluation of this layer was conducted in two phases. The initial phase involved analyzing 130 tweets from the training set used in the preceding layer. The purpose of this was to examine the instances where the model was unsuccessful in predicting the original labels. The second phase was testing the model on 100 tweets from the test data from the previous layer. This aimed to check the predictions of all layers with the original labels and get quantitative results about whether the second layer actually helped in identifying misclassified labels from the first layer or not.

6.2.1 Qualitative Results

After analyzing the 130 tweets from the training set, we have discovered some cases where the model was not able to perform well. Below are some cases where the model failed.

Very Short and General Tweets

Consider tweets like "*Love skiing*", "*I love fruit*", and "*Summer is lovely*" with respective IDs #24, #25, and #33. Such tweets could lead to retrieving a Reddit page or similar that includes these phrases. A stance classification model might classify these as 'agreeing' since the webpage contains all words from the tweet, although these tweets are not related to natural disasters.

Bad Google API Call

Since we use web scraping, results are not always good. Like what happened with the tweet #652, some API calls just returned the content of advertisements, instead of the original content of the page. For tweet #105, the API returned a Captcha as the document. These cases most probably lead the model to incorrectly classify these tweets.

Irrelevant Websites

For some tweets, the retrieved website is totally irrelevant. An example of these websites are StackOverflow, Medium, and Serato. These are totally irrelevant to natural disasters, however, a tweet can match their content normally. For the stance classifier, it will return 1 if the document matches, even if the document is irrelevant to our domain of natural disasters.

6.2.2 Quantitative Results

Upon examining the tweets from our training set, we pinpointed instances where the model did not perform as expected. Utilizing these insights, we eliminated certain flawed outcomes, such as those resulting from bad API calls. Consequently, this process resulted in a refined set of 100 tweets from our test dataset, which we then utilized to obtain quantifiable results. For all results, it's clear that balancing the dataset made a significant difference.

We evaluated the output of the second layer using three different methods. Firstly, we assessed its accuracy in comparison to the actual labels in Table 2. Secondly, we measured how our model performed when the first layer correctly classified the tweet 3. Finally, in Table 4 we examined how well the second layer performed in instances where the first layer had incorrectly classified tweets, comparing this output against the true labels.

	Accuracy	Precision	Recall	F1-Score
Balanced	0.77	0.80	0.73	0.76
Unbalanced	0.69	0.75	0.59	0.66

Table 2: Results for Layer 2 Compared to Actual Labels

Table 2 shows that the model trained with a balanced dataset achieved a 77% accuracy against true labels, successfully verifying the stance for 77 out of 100 tweets.

	Accuracy	Precision	Recall	F1-Score
Balanced	0.77	0.75	0.77	0.76
Unbalanced	0.70	0.60	0.73	0.66

Table 3: Results for Layer 2 Given True Classification From the First Layer

When the natural disaster model accurately identified a tweet, the second layer also demonstrated a 77% accuracy rate in alignment with the true label, as shown in 3. This indicates that it was successful in correctly verifying 77% of the tweets it processed.

In instances where the first layer incorrectly classified a tweet, the second layer accurately identified the correct label in 76% of these cases 4. This finding is particularly significant as it demonstrates the second layer's effectiveness in correcting misclassifications made by the first layer.

	Accuracy	Precision	Recall	F1-Score
Balanced	0.76	0.64	1.00	0.78
Unbalanced	0.65	0.55	0.86	0.67

Table 4: Results for Layer 2 Given False Classification From the First Layer

7 Limitations

After analyzing the outputs manually, we have discovered some flaws in our method. First of all, the size of the dataset for the first layer is small. We tried to solve this issue using data augmentation, however, we believe a bigger dataset would be more reliable. Another issue was the length of the tweets. As shown in the qualitative results, many tweets were very short and general, which makes it more challenging for both layers to classify. Moreover, lots of tweets were missing the keyword and location features. This would have helped a lot, especially in the second layer, where we can verify a tweet according to the location in addition to the normal stance classification method. The dataset also did not include the time when this tweet was posted. This would have been also very helpful in the second layer when we retrieved the documents.

In developing the second layer, we faced limitations with the initial dataset, as it didn't align with our specific needs. Therefore, we fine-tuned the model using a different, more structured dataset, not sourced from Twitter, to better suit our task requirements. The choice of using Google Custom Search API for the second layer turned out to be less effective due to its limitations. The primary issues were the restricted number of API calls, capped at 100 per day, and a high incidence of errors or empty results from these calls. This prevented us from collecting documents for all of the tweets in the original dataset, so we only opted for 100 tweets to test this approach with. Lots of the retrieved documents included errors or missed the actual content. We believe that if we were able to retrieve better-quality documents for each tweet, we would have much better results.

8 Conclusion

The issue of fake news detection has become increasingly critical in the current era, particularly with the rising usage of social media platforms. These platforms allow virtually anyone to publish content, which has led to a significant increase in the spread of misinformation and fake news. To solve this issue, we aimed to create a dual-layered approach to classify and verify that a certain tweet is a natural disaster, thereby enhancing the accuracy and reliability of information identified and disseminated on these platforms.

For the natural disaster classification problem, our proposed model achieved a public F1 score of 0.82653 as our best score. This is around 4.9% higher than the baseline model, which only achieved a 0.77719 score. The improvement in our model's performance was primarily driven by an enhanced methodology that combines BerTopic for feature extraction and RoBERTa for classification. This approach, integrating advanced techniques for both extracting relevant features and performing classification tasks, contributed significantly to the model's enhanced effectiveness. For the stance classification problem in the second layer, the model showed a 77% agreement percentage of the correctly classified tweets from the previous layer. Most importantly, it showed that 76% of the tweets that were misclassified by the first layer were correctly classified in the second layer.

Enhancing our model further can be achieved by integrating prior knowledge, particularly in recognizing areas with varying susceptibility to natural disasters. This can be facilitated by using Large Language Models (LLMs) to extract and analyze statistical data. The benefits of incorporating such prior knowledge are manifold. It can lead to greater accuracy, more efficient handling of sparse data, quicker model convergence, and enhanced regularization. Additionally, it provides domain-specific insights, aids in the integration of uncertainties, promotes better generalization, increases resource efficiency, and aligns with a principle-based modeling approach. As a future step in our model's evolution, we may be able to improve prediction capabilities by utilizing these priors.

References

[1] The characteristics of rumor spreaders on twitter: A quantitative analysis on real data. *Computer Communications*, 160:674–687, 2020.

[2] Phil Culliton Yufeng Guo Addison Howard, devrishi. Natural language processing with disaster tweets, 2019.

[3] Fatima Alhaj, Ali Al-Haj, Ahmad Sharieh, and Riad Jabri. Improving arabic cognitive distortion classification in twitter using bertopic. *International Journal of Advanced Computer Science and Applications*, 13(1):854–860, 2022.

[4] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36, May 2017.

[5] Shahzad Ashraf, Sehrish Saleem, Tauqeer Ahmed, Zeeshan Aslam, and Durr Muhammad. Conversion of adverse data corpus to shrewd output using sampling metrics. *Visual Computing for Industry, Biomedicine, and Art*, 3, 2020.

[6] Mohamed Barbouch, Frank W. Takes, and S. Verberne. Combining language models and network features for relevance-based tweet classification. pages 15–27, 2020.

[7] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. A survey on data augmentation for text classification. *ACM Computing Surveys*, 55(7):1–39, 2022.

[8] Cody Buntain and Jennifer Golbeck. Automatically identifying fake news in popular twitter threads. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 208–215, 2017.

[9] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web*, pages 675–684, 03 2011.

[10] Sajjad Dadkhah, Xichen Zhang, Alexander Gerald Weismann, Amir Firouzi, and Ali A. Ghorbani. The largest social media ground-truth dataset for real/fake content: Truthseeker. *IEEE Transactions on Computational Social Systems*, pages 1–15, 2023.

[11] William Ferreira and Andreas Vlachos. Emergent: a novel data-set for stance classification. 03 2016.

[12] Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. A retrospective analysis of the fake news challenge stance detection task, 2018.

[13] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Processing social media messages in mass emergency: A survey. *ACM Comput. Surv.*, 47(4), jun 2015.

[14] Muhammad Imran, Prasenjit Mitra, and Jaideep Srivastava. Cross-language domain adaptation for classifying crisis-related short messages. 02 2016.

[15] Omid Kashefi and Rebecca Hwa. Quantifying the evaluation of heuristic methods for textual data augmentation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 200–208, 2020.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Bernardo Magnini, Roberto Zanoli, Ido Dagan, Kathrin Eichler, Guenter Neumann, Tae-Gil Noh, Sebastian Pado, Asher Stern, and Omer Levy. The excitement open platform for textual inferences. In Kalina Bontcheva and Jingbo Zhu, editors, *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 43–48, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[18] Martin Müller, Marcel Salathé, and Per E Kummervold. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *Frontiers in Artificial Intelligence*, 6:1023281, 2023.

[19] C. Silverman and Others. Lies, damn lies, and viral content: How news websites spread (and debunk) online rumors, unverified claims, and misinformation. *arXiv Preprint*, 2016.

[20] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*, 2018.

[21] Gangloff A. Schwarz L. Tuech, J. J. Our challenge is to adapt the organization of our system to the six stages of the epidemic to go beyond the covid-19 crisis. *The British journal of surgery, 107(7), e189.*, 02 2020.

[22] Kaggle User. Getting started with pytorch - roberta. https://www.kaggle.com/code/zaber666/getting-started-with-pytorch-roberta, 2023.

[23] S. Vosoughi and Others. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[24] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.