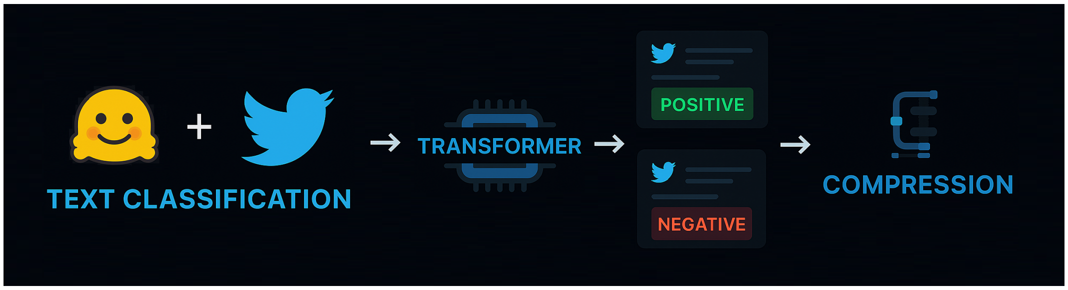
Advanced Topics in Deep Learning - Project Report - COVID-19 Tweet Sentiment Classification Using Transformer-Based NLP Models

Idan Kanat Ido Shahar

August 21, 2025

GitHub Repository



Abstract

The COVID-19 pandemic highlighted the critical role of natural language processing (NLP) in analyzing large-scale textual data from social media to track public opinion and sentiment. In this project, we addressed the task of sentiment classification using the publicly available COVID-19 Twitter NLP Kaggle dataset, which contains over 40,000 labeled tweets. Our goal was to investigate state-of-the-art transformer-based models and evaluate their performance under different fine-tuning and optimization strategies. Specifically, we employed two domain-relevant architectures, BERTweet-Base and Twitter-RoBERTa-Base, which we fine-tuned using both Hugging Face's Trainer API and a custom PyTorch training loop. We conducted systematic hyperparameter optimization with Optuna, accompanied by final training and experiment tracking using Weights & Biases . To reduce model size and improve efficiency, we further applied three compression techniques — quantization, pruning, and knowledge distillation, and analyzed the resulting trade-offs between accuracy and computational efficiency. Our experimental results demonstrate the effectiveness of transfer learning for social media sentiment analysis, while also highlighting trade-offs between model accuracy, training complexity, and inference efficiency. The findings provide insights into practical considerations for deploying NLP models in resource-constrained settings.

Introduction

The COVID-19 pandemic generated unprecedented volumes of online discourse, particularly on Twitter, which quickly became a primary outlet for public opinion, news sharing, and emotional expression. Understanding such sentiment is critical for policymakers and researchers aiming to monitor public response and combat misinformation during health crises. Large-scale datasets such as the COVID-19 Twitter corpus by Chen et al. [1] have enabled this line of work, while Jelodar et al. [2] demonstrated deep sentiment classification and topic modeling on COVID-19 discussions.

Analyzing tweets, however, poses significant challenges due to their short length, informal style, and frequent use of domain-specific language. Traditional natural language processing (NLP) approaches often fail in this setting. In contrast, transformer-based language models such as BERT [3], RoBERTa [4], and domain-adapted variants like BERTweet [5] have achieved state-of-the-art performance on text classification and sentiment analysis tasks, including applications to COVID-19 related social media data. Recent surveys, such as Albladi et al. [6], further highlight the promise of these models while underscoring persistent challenges in efficiency, scalability, and resource constraints.

Building on this context, our work addresses the task of sentiment classification using the publicly available Kaggle dataset of over 40,000 COVID-19 related tweets. We fine-tuned two transformer architectures, BERTweet-Base and Twitter-RoBERTa-Base, under different optimization strategies, and systematically evaluated their performance. Our methodology included hyperparameter optimization with Optuna, experiment tracking with Weights & Biases, and model compression through quantization, pruning, and knowledge distillation. Our goal was to investigate trade-offs between predictive accuracy and computational efficiency, offering insights into practical deployment of NLP models in resource-constrained environments.

Part A - Exploratory Data Analysis

The Dataset

In this project, we used the COVID-19 Twitter NLP Kaggle dataset, containing tweets from March to April 2020 - some of the most intense months of the COVID-19 pandemic. The dataset was already split into training (with over 40,000 tweets) and test data. Our exploratory analysis focused solely on the training data, under the assumption that early inspection of these examples would provide insights useful for downstream modeling. Each record included the original tweet content, sentiment label (five classes - Extremely Negative, Negative, Neutral, Positive and Extremely Positive), the user's reported location and tweet's posting date. For future modeling purposes, we primarily focused on the tweet text and sentiment labels, while our exploratory analysis (detailed below) revealed key insights about the additional fields.

Figure 1 shows the sentiment distribution in the training set. The dataset is not perfectly balanced but also not severely skewed: positive tweets are more frequent than negative ones, and the same holds at the extremes, where extremely positive tweets outnumber extremely negative ones. Taken together, the positive classes clearly surpass the negative, with the gap widening once extremes are included.

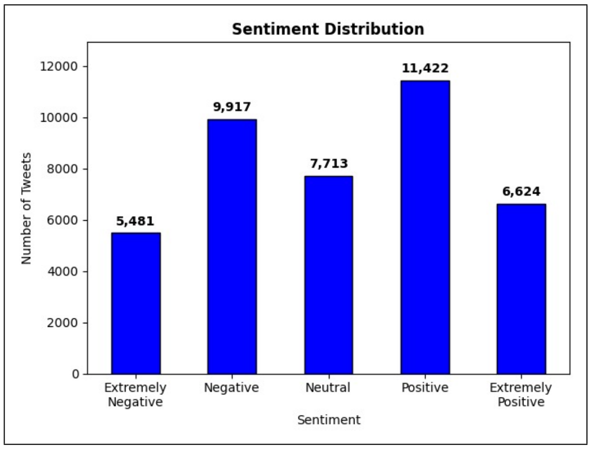


Figure 1: Sentiment label distribution in the training dataset.

We also analyzed tweet length distributions by sentiment, (see Appendix, Figure A1), to inspect potential associations between the two. Overall, lengths are right-skewed with peaks near the 280-character limit, while neutral tweets spread more evenly around 100–200 characters. Polarized sentiments (positive, negative, and extremes) cluster more at longer lengths, suggesting elaboration in stronger opinions.

Temporal and geographic patterns: We examined when and where tweets appeared to contextualize the dataset. Activity spiked in mid-March (19–21 Mar; minor bump 24 Mar) before tapering, with sentiment proportions staying consistent (positive > neutral > extremes). Location mapping showed ∼20% missing, but among valid entries a clear **US dominance (83.8%)** emerged, followed by **UK/Commonwealth (12.7%)** and small shares elsewhere. Overall, the dataset is marked by sharp temporal peaks and US-centric coverage (see Appendix, Figures A2– A3).

Data Cleaning

The raw Corona_NLP dataset contained noisy tweet text, requiring systematic preprocessing to ensure reliable input for sentiment classification. Cleaning steps included expanding English contractions (e.g., *don't* → *do not*), replacing URLs and user mentions with neutral placeholders, and simplifying hashtags while preserving their semantic root. Unnecessary punctuation was stripped, whitespace normalized, and all text converted to lowercase for token consistency. The cleaned versions of the datasets, containing an added `CleanTweet` column, were then saved as new CSV files for further modeling. This pipeline ensured that sentiment cues remained intact while irrelevant noise was removed, facilitating robust model training.

Part B - Fine-Tuning and Modeling Methodology

Data Splitting

To maintain consistency in label distributions, the cleaned training dataset was split into reduced training and validation subsets using **stratified sampling**. This guaranteed balanced representation in both subsets across all 5 sentiment categories:*Extremely Negative*, *Negative*, *Neutral*, *Positive*, *Extremely Positive*.
Label mappings for the sentiments to the integers 0-4 were explicitly defined prior to splitting, ensuring reproducibility and alignment with model expectations.
The resulting partition comprised approximately 83% training, 8.5% validation, and 8.5% test data. Validation and test sizes were kept identical by design to support fair comparisons and simplify performance tracking. This strategy provided a rigorous evaluation setup: the training subset for learning, the validation subset for hyperparameter tuning and performance monitoring, and the untouched test set for final model assessment. We considered K-Fold cross-validation for more robust metric estimation but opted for the proposed simpler split due to computational & time constraints.

Modeling Methodology

We trained and evaluated two transformer-based architectures from HuggingFace — **RoBERTa-Base-Tweet** (Model 1) and **BERTweet-Base** (Model 2), on the `Corona_NLP` dataset. Both models were chosen as robust, domain-adapted baselines tailored to the unique linguistic nuances of Twitter data. RoBERTa-Base-Tweet was adapted from RoBERTa with additional pretraining on large-scale (∼58M) tweets, while BERTweet is a RoBERTa-based model trained from scratch on ∼850M tweets, making both of them particularly effective for capturing Twitter-specific language patterns. Our evaluation focused on the **accuracy** metric, measured on the ”unseen” validation set, chosen given the relatively balanced distribution of the five sentiment classes.
For each model, we implemented two training methods:
Custom Loop: A manual implementation, we iterated over batches using PyTorch, applied mixed precision training, and manually computed evaluation metrics. This setup allowed explicit integration of early stopping with patience. The loop also provided flexibility in logging additional statistics.
HuggingFace Trainer: The Trainer handled batch scheduling, gradient accumulation, logging, and evaluation automatically. Metrics including accuracy, precision, recall, and F1-score were tracked at the end of each epoch. Best model checkpoints were saved according to **validation accuracy**.

This dual approach allowed us to compare the built-in optimization pipeline with a more flexible loop where we could explicitly control gradient scaling, evaluation frequency, and early stopping.

Training Procedure

Hyperparameter tuning was the first step of our pipeline, carried out with the `Optuna` hyperparameter optimization package to efficiently explore the search space and identify the best-performing trial based on validation accuracy, final training was performed on the combined training and validation sets and evaluated on the held-out test set. In both steps, performance metrics (train, validation & test) were continuously tracked and visualized using the `Weights & Biases (W&B)` API, ensuring systematic monitoring. The hyperparameters explored during training included: a **learning rate** sampled log-uniformly within the range of $[10^{-5}, 10^{-3}]$ and a **weight decay** sampled log-uniformly within $[10^{-6}, 10^{-4}]$. **Batch sizes** were chosen from the discrete set $\{32, 64, 128\}$. To allow flexible fine-tuning, we varied the number of transformer layers to be unfrozen across integers in $[0, 3]$, explicitly allowing the option of no unfreezing, if the pre-trained model delivers competitive performance. Early stopping was applied with a patience of 7–10 epochs without improvement over the best validation accuracy, ensuring stable convergence. Finally, each model was allocated up to 12 trials.

Improvement Measures

To reduce runtime costs, we applied a pre-tokenization step prior to model training. The dataset was preprocessed once with HuggingFace tokenizers, generating fixed-length input sequences and attention masks. These pre-tokenized files were cached and re-used across Optuna trials, significantly improving efficiency. In addition, the number of Optuna trials was incrementally increased from 10 to 12 to balance computational feasibility with sufficient search coverage. As runtime allowed, the maximum number of epochs per trial was also raised, first from 15 to 20 and finally to 25 during the last training phase, in order to encourage more robust convergence. Collectively, these adjustments led to marked improvements in trial performance, with models progressing from near-random baseline behavior to competitive accuracy levels around 0.7.

Model Compression Techniques

To address the substantial computational demands of the fine-tuned transformer-based NLP models, we applied three model compression techniques on all 4 fine-tuned models. Our motivations were twofold: reducing model size to ease deployment and storage, and accelerating inference to meet practical constraints, particularly given limited training time. Importantly, we applied each technique to all trained models, not only the best-performing ones, in order to evaluate whether compressed variants could sometimes perform better than expected.

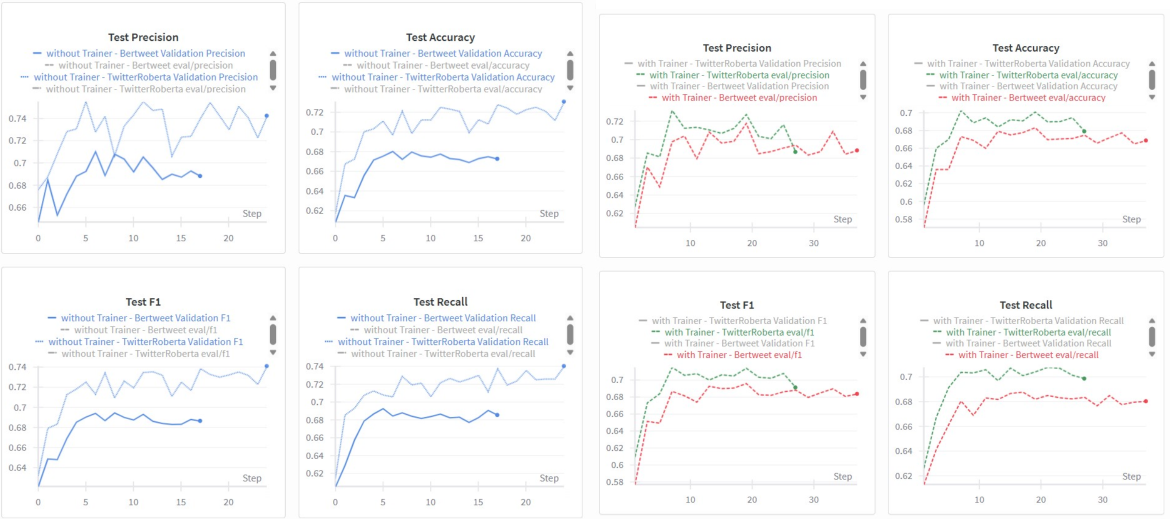
First, we performed **dynamic quantization** (qint8), which reduces the precision of weights to 8-bit integers, significantly decreasing memory footprint and speeding up inference. Second, we applied **unstructured global pruning**, progressively removing low-magnitude weights across either all linear layers or the full model. For this purpose, we implemented a flexible pruning function allowing different granularity levels of weight removal. Finally, we conducted **knowledge distillation (KD)**, where a lightweight RoBERTa variant from Hugging Face was trained for five epochs using logits from the fine-tuned teacher models as soft targets. For each compressed model, we systematically tracked performance metrics, parameter counts, reduction ratios, and drops in accuracy, thereby quantifying both the efficiency gains and the trade-offs in predictive quality.

Results and Conclusions

As shown in Figures 2, 3, generated via W & B, across both models and training methods, we monitored **test** metrics as:

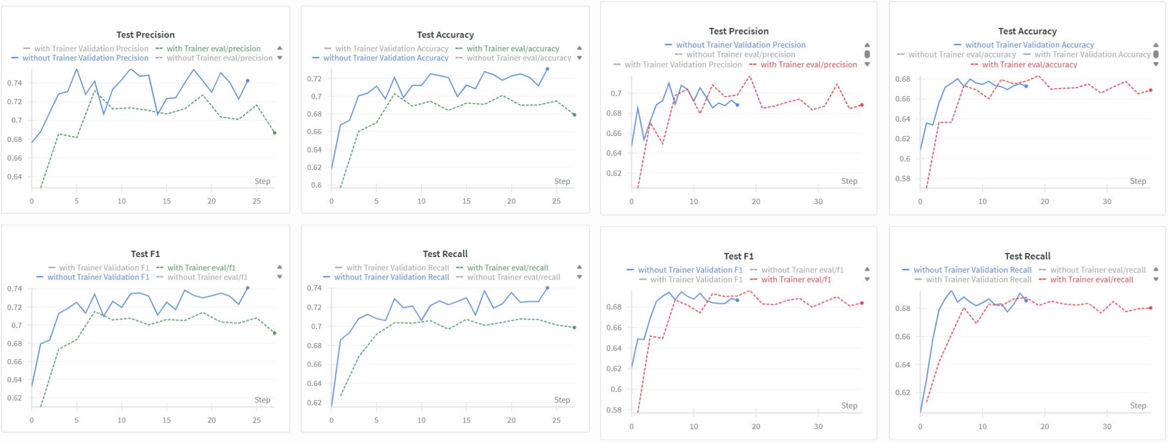
- **Accuracy:** Initialized at around 0.62-0.64 for most models and typically plateaued around 0.68–0.72 on the test set, indicating strong discrimination on unseen data, especially on a difficult 5 class problem.
- **F1-score:** Ranged from 0.58–0.74, showing balanced performance across sentiment classes.
- **Precision/Recall:** both ranged around 0.66–0.74, indicating balanced performance across sentiments without strong bias.

Figure 2: Comparison of the models: Model 1 vs. Model 2.



(a) Model 1 (Dashed) vs. Model 2 (Stark), no Trainer (b) Model 1 (Green) vs. Model 2 (Red), with Trainer

Figure 3: Comparison of the training methods: With vs. Without Trainer.



(a) With (Blue) vs. Without (Green) Trainer, Model 1 (b) With (Blue) vs. Without (Red) Trainer, Model 2

The **TwitterRoBERTa-Base model (1) consistently achieved higher test accuracy, precision, recall, and F1-scores** than the BERTweet-Base model (2) across almost all steps of final training (with test metrics continuously logged), both with and without the HuggingFace Trainer. When using the Trainer, the performance gap between the two models was similar to the no-Trainer setup; TwitterRoBERTa still held a modest edge, though BERTweet tracked closer in this configuration, suggesting that the Trainer reduced disparities while maintaining TwitterRoBERTa's advantage.

Across both models, all four metrics generally reached **higher peaks** in the runs conducted **without the Trainer.** In BERTweet, however, the no-Trainer pipeline triggered early stopping sooner, indicating weaker convergence under this setup.

Importantly, all models exhibited similar **fluctuations** in their performance curves. Even though one training type or model often appeared marginally stronger, the fact that their trajectories remained close (hovering around ∼0.7 for most metrics, as seen in the figures) suggests that their **overall performances were relatively comparable.**

We also compared the compressed model performances across all 4 models and 3 compression techniques (12 compressions in total).*Note:* The original table included a wider set of metrics (train

Table 1: Key Metrics of All Compressed Versions of All Trained Models

Model	Method	Compressed Params	Accuracy (Test)	F1 (Test)
BERTweet-Base (No Trainer)	K.D	20,563,269	0.672	0.685
BERTweet-Base (Trainer)	K.D	20,563,269	0.677	0.689
RoBERTa-Base-Tweet (No Trainer)	K.D	17,046,853	0.688	0.7
RoBERTa-Base-Tweet (Trainer)	K.D	17,046,853	**0.69**	**0.704**
BERTweet-Base (No Trainer)	Pruning	80,983,455	0.538	0.513
BERTweet-Base (Trainer)	Pruning	80,983,455	0.541	0.516
RoBERTa-Base-Tweet (No Trainer)	Pruning	74,830,853	0.517	0.508
RoBERTa-Base-Tweet (Trainer)	Pruning	74,830,853	0.198	0.107
BERTweet-Base (No Trainer)	Quantization	49,291,776	0.162	0.056
BERTweet-Base (Trainer)	Quantization	49,291,776	0.233	0.154
RoBERTa-Base-Tweet (No Trainer)	Quantization	39,037,440	0.664	0.668
RoBERTa-Base-Tweet (Trainer)	Quantization	39,037,440	0.448	0.353

& test), but due to space constraints we report only the key ones: **compressed model parameter counts, test accuracy, and F1-scores**. The original parameter counts remained unchanged across models: 124,649,477 for the Twitter-RoBERTa-Base model (1) and 134,903,813 for the BERTweet-Base model (2). *K.D.* denotes Knowledge Distillation in the table.

Overall, Table 1 highlights clear trade-offs among compression methods. **Knowledge Distillation (K.D.) delivered the strongest results**, preserving high accuracy (up to 0.69) and the top F1-score (0.704), similar to the original models' performances, despite the most aggressive parameter reduction (∼17M–20M parameters in the tiny RoBERTa student). **Pruning**, while retaining far more parameters (∼74M–80M), often collapsed performance to ∼0.5 in both accuracy and F1, and even lower for TwitterRoBERTa-Base with the Trainer, making it unreliable. **Quantization** also behaved inconsistently: RoBERTa without Trainer maintained competitive scores (accuracy ∼0.66, F1 ∼0.67) at ∼39M parameters, but both BERTweet-Based models were severely degraded, nearing 5-class random baseline (0.2). Taken together, these results show distillation as the most reliable strategy by far, while quantization and pruning are unstable, with effectiveness highly model-dependent despite their less aggressive reductions.

Limitations and Directions for Extended Experimentation

The most serious challenge in this project was the severe time constraint. Combined with heavy computational demands, this prevented broad trial-and-error experimentation. The only incremental tuning we managed was modest: **increasing Optuna trials from 10 to 12, raising epochs per trial from 15 to 20, and extending the final training from 20 to 25 epochs**. Beyond these adjustments, further experimentation was not feasible within scope.

With more time and resources, we would (i) **expand the hyperparameter search space**, with larger ranges of fine-tuned layers (potentially up to 5), dropout, and 30+ trials for better coverage; (ii) explore richer **compression regimes**, such as structured pruning (removing full heads/layers) and quantization-aware training, to compare with distillation; (iii) adopt **stratified k fold cross-validation** to reduce variance from single splits and better capture generalization and possibly; (iv) conduct systematic **error analysis**, examining per-class confusion and tweet-length effects to guide targeted data augmentation. Together, these steps would deepen insights into robustness, stability under compression, and practical trade-offs for deployment.

In summary, our work: (i) we fine-tuned two HuggingFace transformer models for COVID-19 tweet sentiment classification (**TwitterRoBERTa-Base** and **BERTweet-Base**), confirming that they can successfully discriminate across all 5 classes, with performance plateauing around ∼0.7 across metrics, still leaving room for improvement. (ii) After training, we applied three post-training compression techniques — **quantization, pruning, and knowledge distillation**, where **knowledge distillation stood out**, reducing parameters most aggressively (∼17–20M vs. ∼125–135M) while retaining performance nearly identical to the original models. In contrast, pruning proved unstable and quantization model-dependent, underscoring distillation's success as the most practical compression strategy.

Appendix A: Supplementary Figures

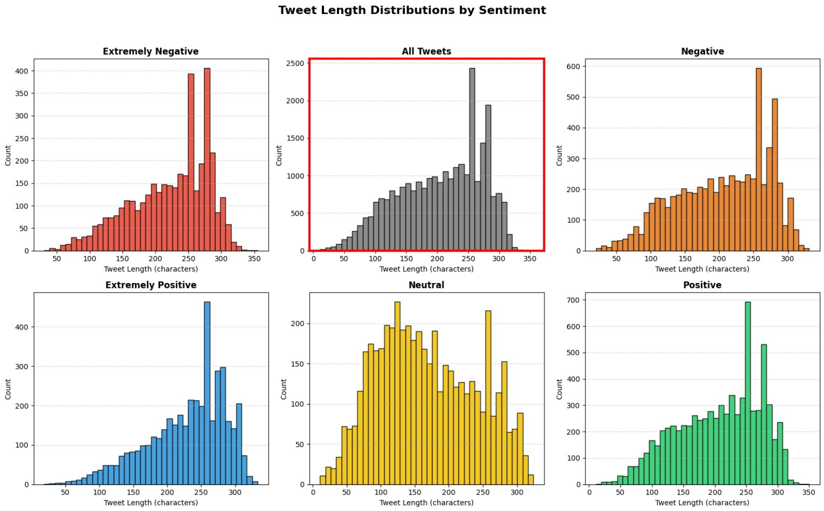


Figure A1: Distribution of tweet lengths stratified by sentiment labels.

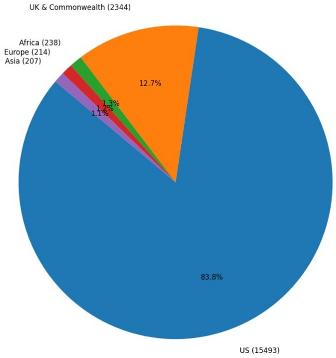


Figure A2: Tweet distribution by region.

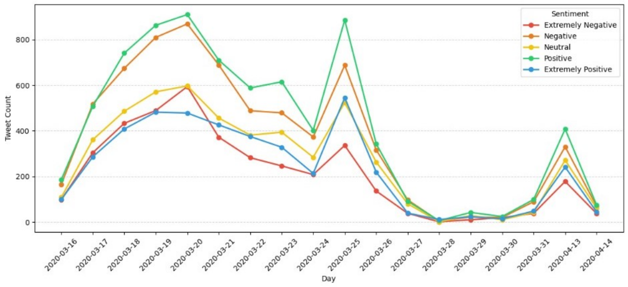


Figure A3: Daily tweet counts by sentiment.

References

[1] Emily Chen, Kristina Lerman, and Emilio Ferrara. Covid-19: The first public coronavirus twitter dataset. *arXiv preprint arXiv:2003.07372*, 2020.

[2] Hamed Jelodar, Yongli Wang, Chi Yuan, Zhiwei Feng, Xiahui Jiang, and Yu Li. Deep sentiment classification and topic discovery on covid-19 online discussions. *arXiv preprint arXiv:2004.11695*, 2020.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[4] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[5] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*, 2020.

[6] Aish Albladi, Minarul Islam, and Cheryl Seals. Sentiment analysis of twitter data using nlp models: A comprehensive review. *IEEE Access*, 13:30444–30468, 2025.