Classifying COVID-19 Tweets Using Deep NLP Models and Compression Methods

Maximiliano Niemetz, Chen Halfi
MSc in Industrial Engineering - Deep Learning course

August 2025

Abstract

This paper explores the use of transformer-based models for sentiment classification of COVID-19 related tweets. We trained and evaluated two models on the Corona NLP dataset, comparing their performance using Hugging Face and PyTorch fine-tuning approaches. In addition, we applied model compression techniques including quantization, pruning, and Half- Precision to reduce model size while preserving accuracy. Our findings demonstrate the trade-off between performance and computational efficiency, with implications for deploying NLP models in resource-constrained environments.

1 Introduction

The COVID-19 pandemic strained healthcare systems worldwide and generated an unprecedented volume of communication on social media, particularly Twitter. Analyzing such content is essential for understanding public sentiment, tracking misinformation, and supporting policy decisions. Sentiment analysis of COVID-19 tweets therefore represents an important NLP challenge.

Traditional machine learning methods such as SVM often struggle with short, noisy text, whereas transformer-based models achieve state-of-the-art performance by leveraging contextual embeddings. In this project, we fine-tune two pretrained transformers on the Corona NLP dataset: **Twitter-RoBERTa**, designed for social media text, and **ELECTRA-base**, which uses a more efficient replaced-token detection objective. Both are evaluated with Accuracy and F1 score.

Beyond model comparison, we analyze the dataset, apply two fine-tuning strategies (a custom PyTorch loop and the Hugging Face Trainer), and experiment with model compression techniques (quantization, pruning, FP16) to balance performance and efficiency. Together, these steps provide insights into the challenges of COVID-19 tweet classification, the relative strengths of Twitter-RoBERTa and ELECTRA, and practical considerations for efficient deployment.

2 Related Work

Twitter has become a key source for sentiment analysis due to its real-time reflection of public opinion. Early studies used classical machine learning methods such as Logistic Regression and SVM with Bag-of-Words or TF-IDF features. Vanga et al. [5] found Logistic Regression with TF-IDF most effective, while AminiMotlagh et al. [1] showed SVM performed best on multi-class Twitter datasets.

Recent work has shifted toward deep learning and transformer-based models. Alam et al. [4] demonstrated that BERT outperforms classical approaches for COVID-19 tweets. Variants like RoBERTa and ELECTRA further improve results, with ELECTRA's replaced token detection offering greater sample efficiency [2].

Motivated by these advances, our project evaluates Twitter-RoBERTa and ELECTRA-base to compare different pretraining strategies (masking vs. replaced token detection) for COVID-19 tweet sentiment classification.

2.1 Exploration of the Data

2.2 The Dataset

We utilized the Corona NLP dataset from Kaggle, a collection of approximately 50,000 tweets posted between March and April 2020, during an intense period of the COVID-19 pandemic. Each tweet is labeled with a sentiment: Extremely Negative (13.3%), Negative (16.1%), Neutral (18.7%), Positive (24.1%), and Extremely Positive (27.8%). The dataset class imbalance was a key consideration in our methodology.

2.3 Exploration of the Data

We began by validating the sentiment distribution of the dataset (Figure 1), which highlighted class imbalance. To address this, we adopted F1-macro as our primary evaluation metric, since it provides a more reliable measure than accuracy in imbalanced settings. All dataset splits (training, validation, and testing) were stratified to preserve this distribution.

Analysis of hashtags showed that most were generic (e.g., #covid19), but some carried emotional content and could enrich sentiment prediction (Figure 6). Compound terms (e.g., #thankyouhealthcareworkers) were split, and hashtags were incorporated as part of the textual input. This treatment allowed us to retain potentially valuable semantic signals while keeping preprocessing consistent.

Our text-cleaning pipeline focused on removing irrelevant tokens (e.g., stop words) and carefully handling tokenization to avoid losing meaning (e.g., "coronavirus" incorrectly truncated to "coronaviru"). Tweets with fewer than five words after cleaning were removed, as they were unlikely to provide meaningful sentiment context. After preprocessing, the training set contained 40,126 tweets. These steps confirmed that focusing on high-quality cleaned text, enriched with hashtags, was more effective than relying on external features such as user location.

We also explored several additional methods, though they proved less informative for this task:

- Temporal analysis revealed two distinct "waves" of tweet activity, but sentiment distributions remained consistent across them (Figures 2, 3).
- Location data, even after cleaning and standardization, showed no significant sentiment differences across countries (Figures 4, 5).
- Tweet length, whether measured in characters or words, showed no correlation with sentiment categories.
- Principal Component Analysis (PCA) failed to separate sentiments into clear or useful clusters (Figure 7).

Overall, these findings reinforced our decision to focus on transformer-based text classification using cleaned tweets and hashtags, while discarding auxiliary features that added complexity without performance gains.

3 Proposed Models

3.1 Model Selection

To compare different transformer paradigms, we selected **RoBERTa** and **ELECTRA**, focusing on how pretraining objectives impact tweet sentiment classification.

RoBERTa [3] extends BERT with masked language modeling (MLM). We first used `roberta-base`, but later adopted **Twitter-RoBERTa**, pretrained on large-scale tweets, to better capture social media language and improve accuracy on the Corona NLP dataset.

ELECTRA [2] employs *replaced token detection*, training the model to spot tokens substituted by a generator. This method is more sample-efficient than MLM and often achieves strong results with fewer resources.

Thus, our study contrasts Twitter-RoBERTa (MLM-based) and ELECTRA-base (replaced token detection) to evaluate their effectiveness for COVID-19 tweet sentiment classification.

3.2 Model Compression Techniques

To address the computational cost of transformer models, we evaluated three widely used compression strategies. **Quantization** reduces model size and inference time by representing weights with lower precision (e.g., 8-bit instead of 32-bit). **Pruning** removes less important parameters from the network, leading to a sparser model without major accuracy loss. Finally, **FP16** half-precision training decreases memory consumption and accelerates computation on modern GPUs. These techniques allow us to examine the trade-off between efficiency and predictive performance when deploying transformer-based sentiment classifiers.

4 Results

4.1 Comparison Between Models

We evaluate two architectures (**Twitter-RoBERTa** and **ELECTRA-base**) under two training frameworks: a manual *full-code* PyTorch loop and the Hugging Face *Trainer* (*HF code*). We report validation metrics (Accuracy, F1, Precision, Recall) to compare both architecture choice and training framework effects.

Table 1: Validation performance of the chosen models and training frameworks. Abbreviations: RoB-Tw = Twitter-RoBERTa, Elec = ELECTRA, Full = Full Code, HF = Hugging Face Trainer.

Metric	**RoB-Tw (Full)**	**RoB-Tw (HF)**	**Elec (Full)**	**Elec (HF)**
Val F1	0.713	0.733	0.723	0.731
Val Accuracy	0.715	0.725	0.724	0.722
Val Precision	0.719	0.728	0.726	0.724
Val Recall	0.715	0.725	0.724	0.723

As shown in Table 1, we observe the relative gains from Twitter pretraining and the consistency between the two training frameworks. In the next subsection we quantify the trade-offs introduced by compression.

During experimentation with Roberta, we first fine-tuned the general-purpose RoBERTa-base model. While it achieved reasonable performance, its accuracy on Twitter data was lower than expected. This motivated us to explore Twitter-RoBERTa, a variant specifically trained on large-scale Twitter corpora. The results confirmed that domain-specific pretraining yields superior performance for noisy, short texts.

We applied Optuna for hyperparameter tuning in order to systematically explore the effect of training configurations. In the initial search space, we focused on the core parameters: *learning rate*, *max sequence length*, *batch size*, *patience*, *number of epochs*, and *weight decay*. After preliminary runs, we extended the search space with additional parameters, including *warmup ratio*, *dropout*, and the choice of *learning rate scheduler*, to further improve results.

For each model, we performed a two-stage tuning strategy. The first run used a broader hyperparameter space to identify promising regions. Based on these results, we conducted a second run with a narrower search space, which allowed for more fine-grained optimization within computational limits. This iterative process helped us balance exploration of diverse configurations with efficient use of resources.

4.2 Comparison of Compressed Models

To reduce inference cost while maintaining accuracy, we apply *post-training quantization*, *pruning*, and *FP16* to the best-performing fine-tuned model. We compare them to the best full-precision baseline (FP32) on the held-out test set.

Table 2: Test performance after compression compared to the best Roberta HF model.

Metric	**Quantization**	**Pruning**	**FP16**	**Best Rob-HF (FP32)**
Test F1	0.688	0.541	0.691	0.691
Test Accuracy	0.688	0.575	0.690	0.690
Test Precision	0.691	0.673	0.693	0.693
Test Recall	0.688	0.575	0.690	0.690

Table 3: Test performance after compression compared to the best Roberta Full model.

Metric	**Quantization**	**Pruning**	**FP16**	**Best Rob-Full Code (FP32)**
Test F1	0.679	0.654	0.677	0.677
Test Accuracy	0.681	0.658	0.679	0.679
Test Precision	0.684	0.702	0.681	0.681
Test Recall	0.681	0.658	0.679	0.679

Table 4: Test performance after compression compared to the best Electra full code.

Metric	**Quantization**	**Pruning**	**FP16**	**Best Rob-Full Code (FP32)**
Test F1	0.338	0.67	0.66	0.66
Test Accuracy	0.406	0.668	0.664	0.665
Test Precision	0.406	0.668	0.664	0.664
Test Recall	0.620	0.689	0.666	0.666

Table 5: Test performance after compression compared to the best Electra HF model.

Metric	**Quantization**	**Pruning**	**FP16**	**Best Rob-Full Code (FP32)**
Test F1	0.467	0.58	0.693	0.693
Test Accuracy	0.503	0.596	0.694	0.693
Test Precision	0.503	0.595	0.693	0.693
Test Recall	0.655	0.675	0.694	0.694

5 Discussion

Our findings demonstrate that Twitter-RoBERTa is superior for this task, largely due to its domain-specific pretraining. This confirms that for niche domains like social media, a tailored model is more effective than a general-purpose one.

The Hugging Face Trainer proved to be a practical and efficient tool, simplifying the fine-tuning process without sacrificing performance.

For real-world deployment, FP16 half-precision training is an excellent choice, as it reduces memory and speeds up computation with virtually no impact on accuracy. Quantization and pruning, while effective for size reduction, resulted in a more noticeable performance drop, suggesting a greater trade-off for these methods.

6 Challenges and Limitations

Throughout the project, we encountered several challenges related to data preprocessing, model training, and experiment management. These difficulties shaped our decisions and highlight practical considerations for future work.

6.1 Integration of Tools

One of the main challenges arose when integrating `Optuna` for hyperparameter tuning with the Hugging Face `Trainer` and logging results to Weights & Biases (W&B). Although the values for hyperparameters such as `max_length` were correctly sampled by Optuna (verified by printed statements inside the code), in the W&B dashboard the parameter was consistently displayed as 20. Despite attempts to debug, this inconsistency remained unresolved and complicated the interpretation of logged experiments.

6.2 CUDA Runtime Errors

During the optimization process with Optuna, we also encountered a recurrent CUDA error (`RuntimeError: CUDA error: device-side assert triggered`). This error originated from Hugging Face's attention mask utilities and was difficult to trace due to asynchronous CUDA kernel reporting. The first approach was to check whether the labels we used were in the correct range (between 0-4), or the type was wrong. After checking those factors and proving that there was no issue, we investigate more, and found that this problem can arise from the fact that we added tokens to the tokenizer (`<url>`, `<user>`, `<number>`) but we didn't resize the model's embedding matrix. To resolve this, it was necessary to explicitly call `model.resize_token_embeddings(len(tokenizer))` to ensure the new tokens were correctly mapped to trainable embeddings. This adjustment was crucial for successful training.

6.3 Computational Constraints

As experiments were run on Google Colab Pro, runtime limits and GPU availability sometimes interrupted long fine-tuning jobs. This required rerunning certain trials and reducing the Optuna search space to make hyperparameter optimization feasible within the available resources. On one occasion, toward the end of a study, the disk space on Colab reached its limit, which forced us to stop the study prematurely.

7 Conclusions

This project explored transformer-based models for COVID-19 tweet sentiment analysis, comparing Twitter-RoBERTa and ELECTRA, and evaluating compression techniques. Results showed that domain-specific pretraining improves accuracy, while efficiency-oriented models and compression can reduce resource demands with limited performance loss.

The main limitation was the restricted time and resources, which narrowed our hyperparameter search and experimental scope. Future work should include more extensive HP tuning, cross-validation, and models combination.

References

[1] Masoud AminiMotlagh, HadiShahriar Shahhoseini, and Nina Fatehi. A reliable sentiment analysis for classification of tweets in social networks. *Social Network Analysis and Mining*, 13(7), 2023.

[2] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*, 2020.

[3] 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.

[4] Aslam Rupapara Mehmood Rustam, Khalid and SangChoi. A performance comparison of supervised machine learning models for covid-19 tweets sentiment analysis. *PLOS ONE*, 16(2):e0245909, 2021.

[5] Kumar Vanga et al. Machine learning models for sentiment analysis of tweets: Comparisons and evaluations. *Machine Learning with Applications*, 7:100198, 2022.

Project GitHub Repository

A EDA Graphs

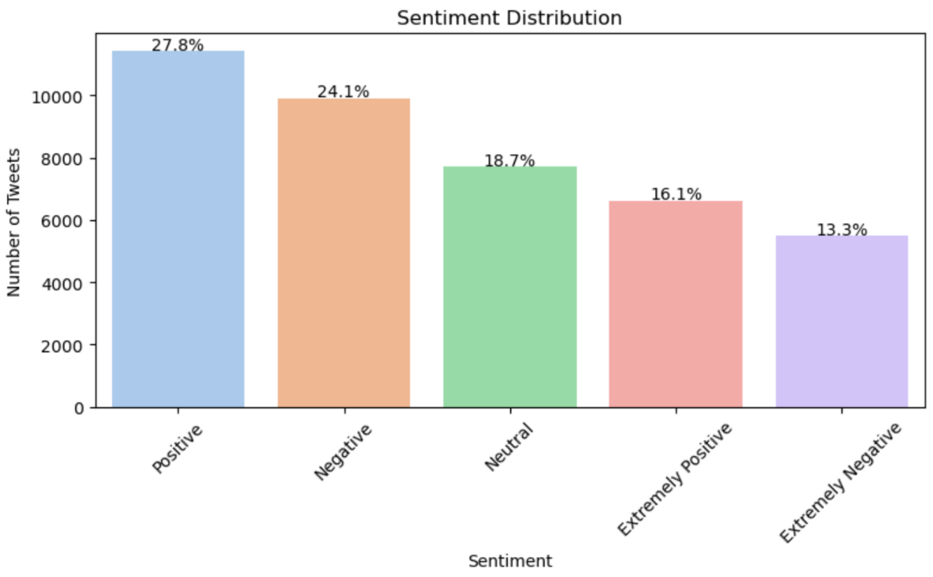


Figure 1: Sentiment Distribution

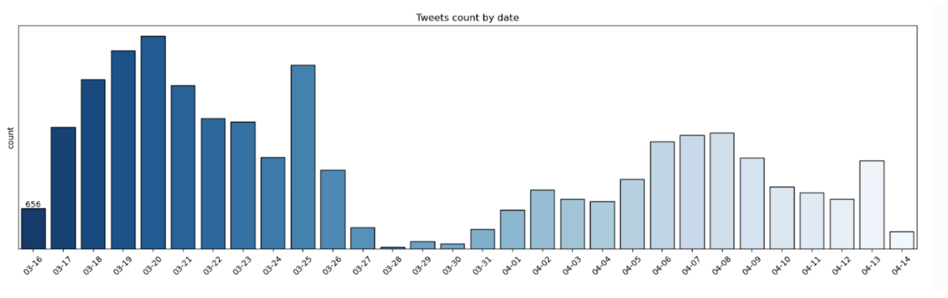


Figure 2: Tweets by date

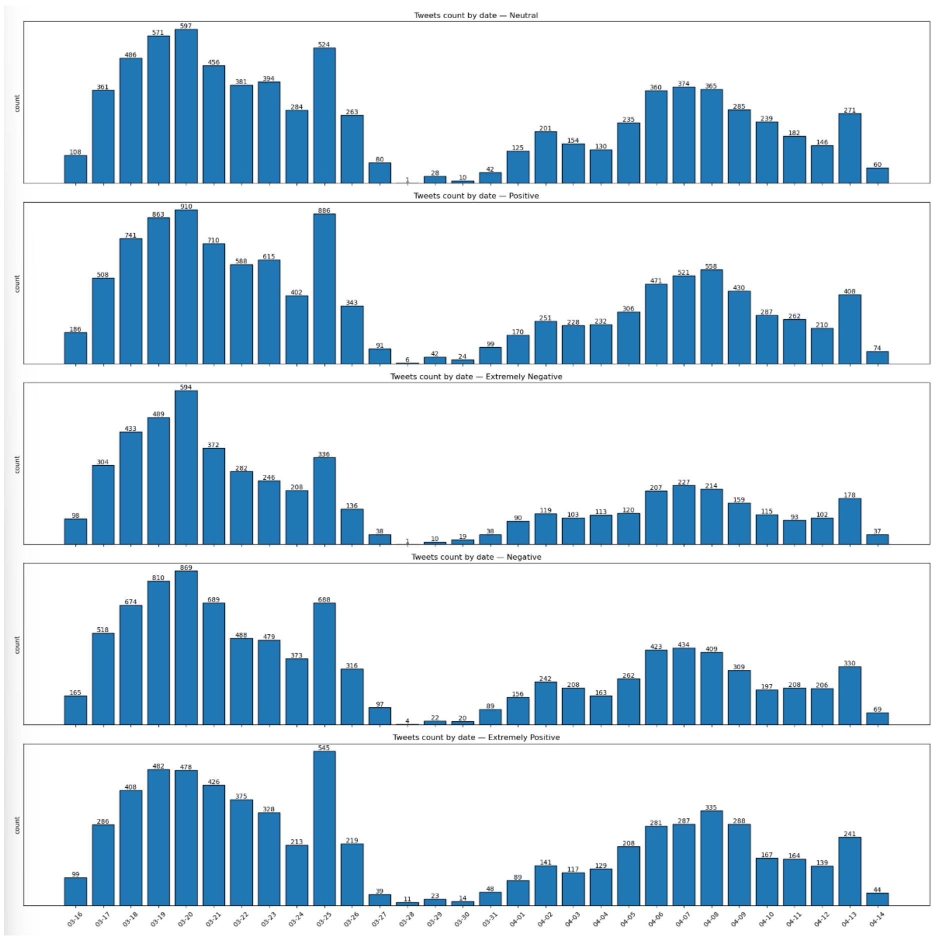
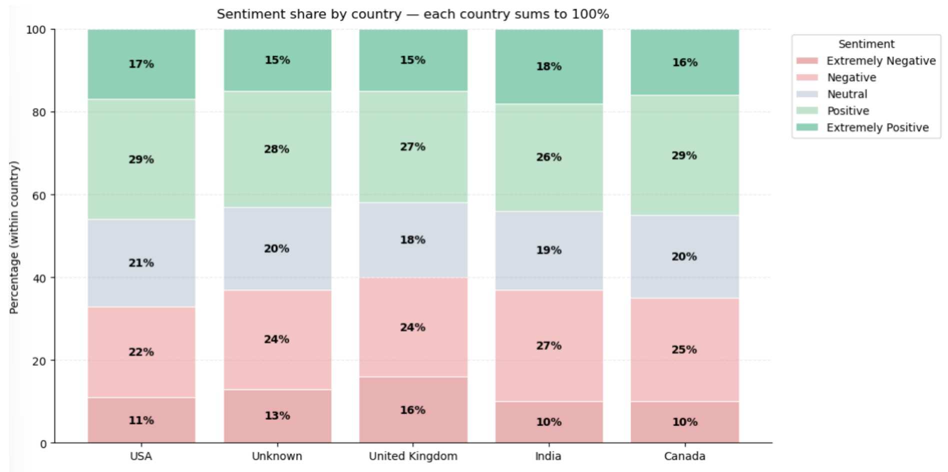


Figure 3: sentiment by dates



The sentiment split looks broadly similar across countries—no country shows a strong unique skew.

Figure 4: sentiment by country

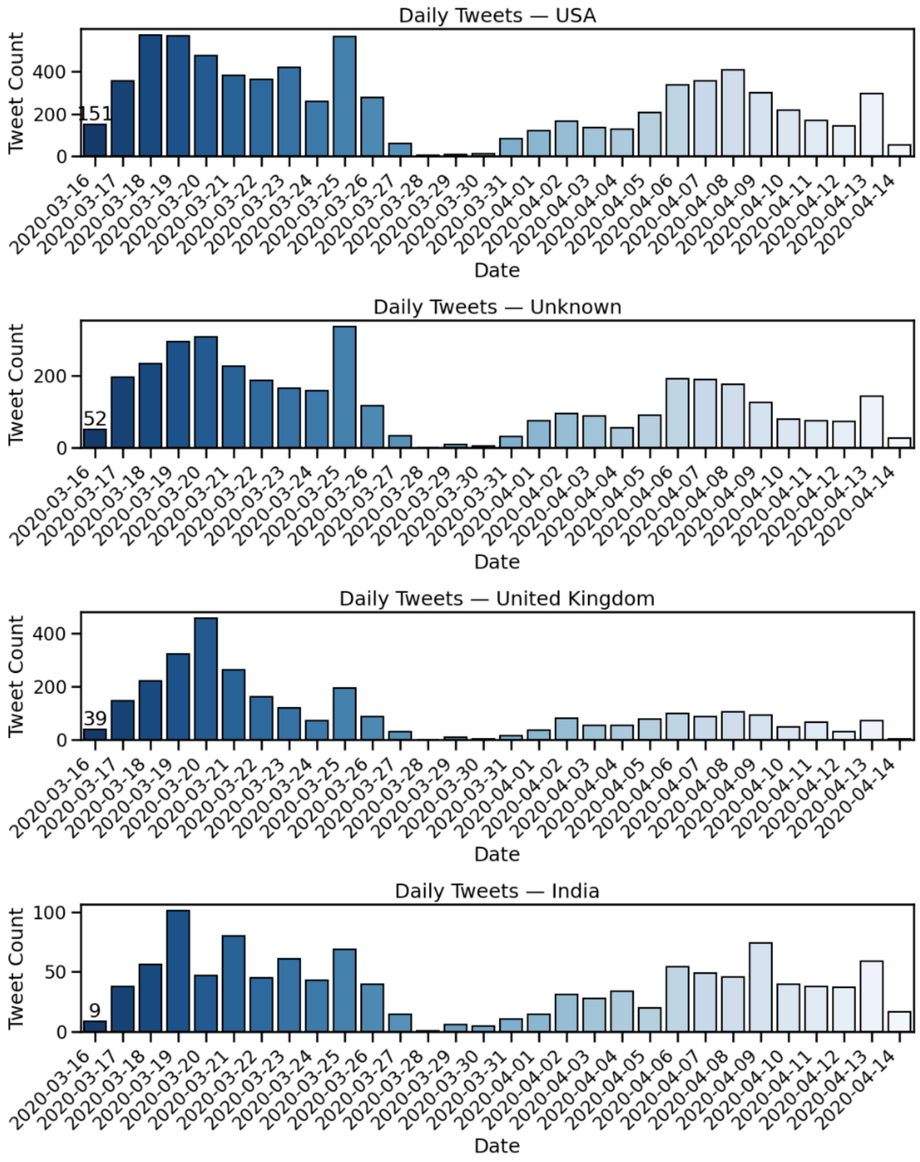


Figure 5: tweets by dates and location

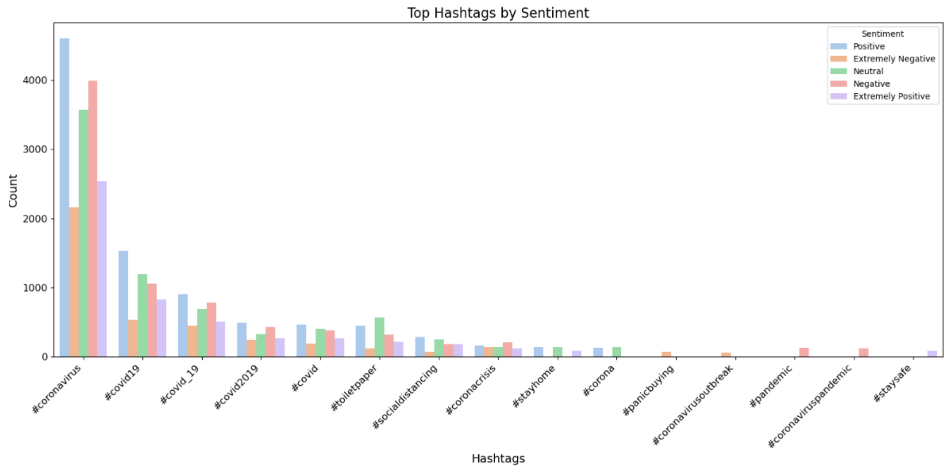


Figure 6: Hashtags

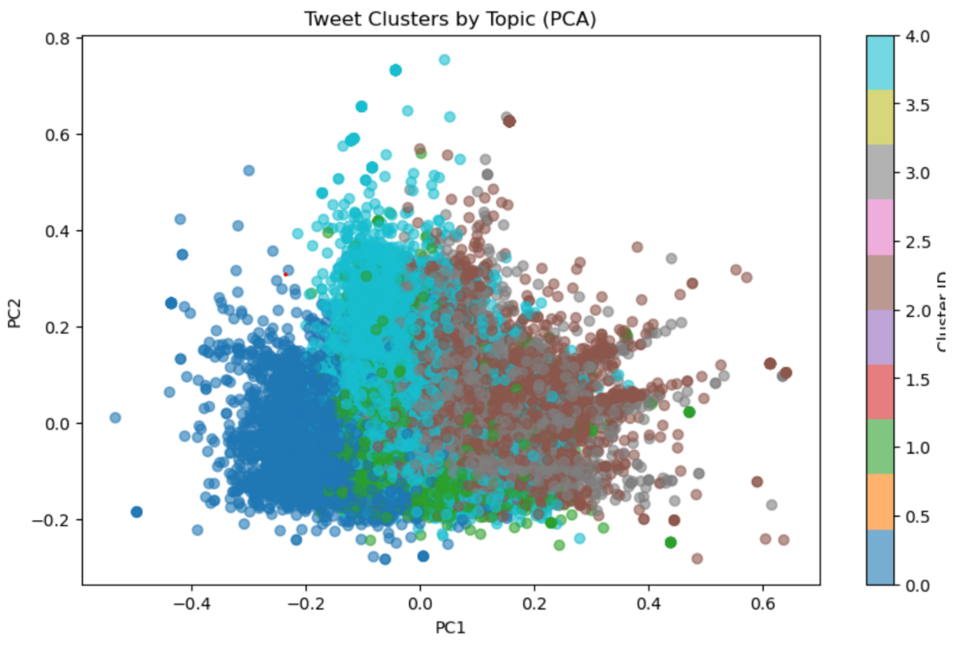


Figure 7: PCA

B Wandb graphs

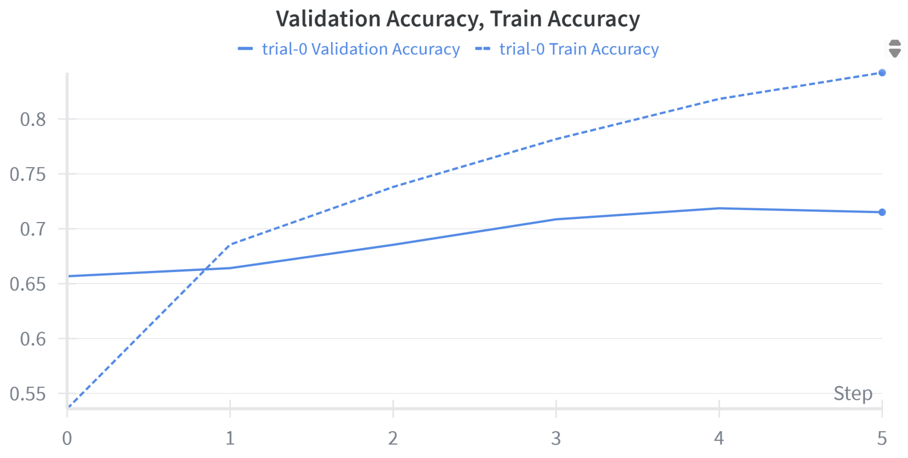


Figure 8: Accuracy for Roberta using Full Code.

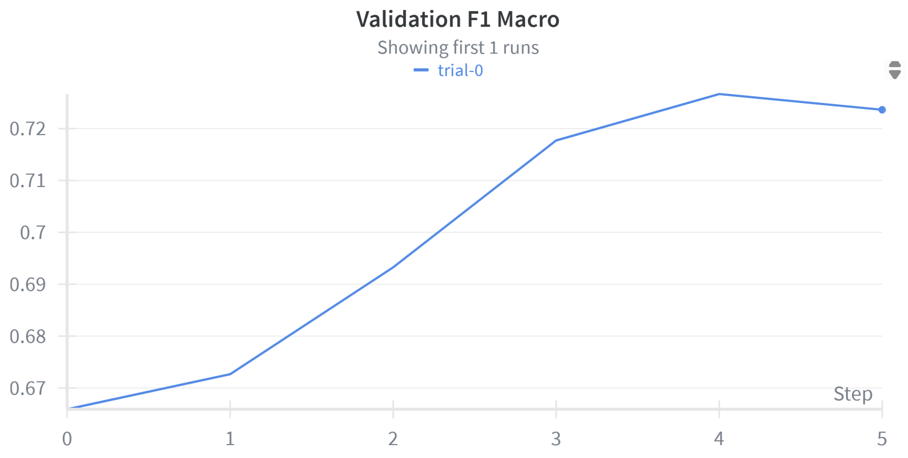


Figure 9: F1 for Roberta using Full Code.

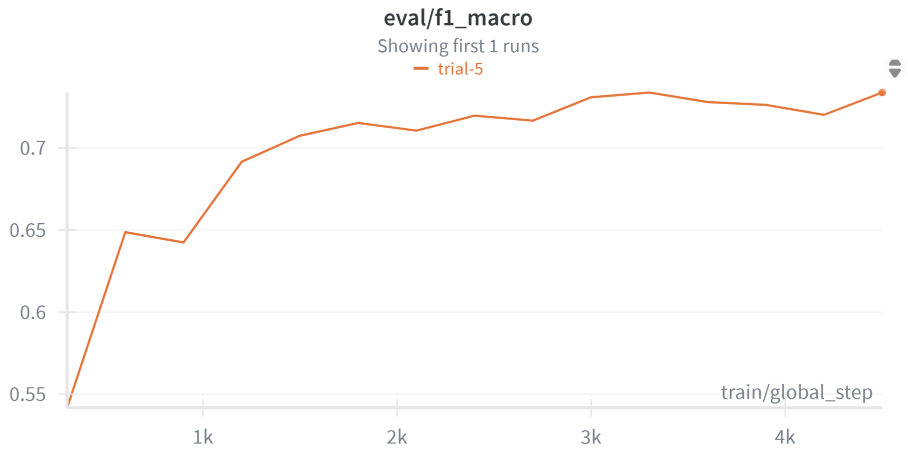


Figure 10: F1 for Roberta using HF Code.

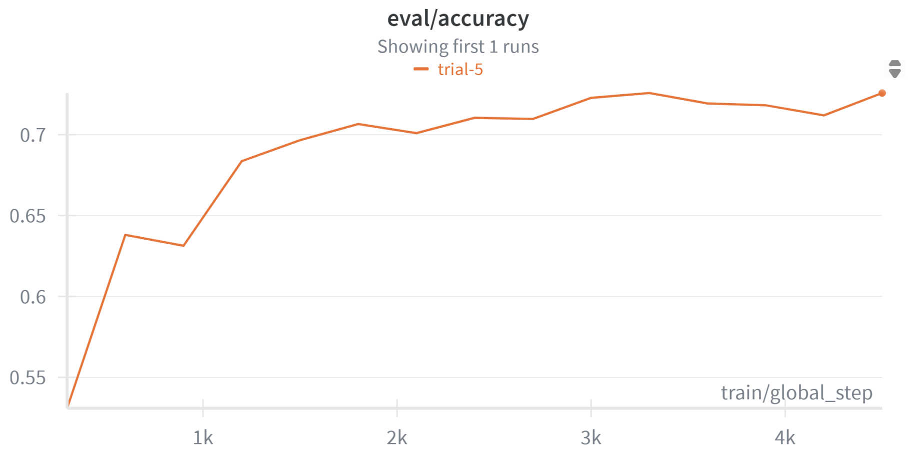


Figure 11: Accuracy for Roberta using HF Code.

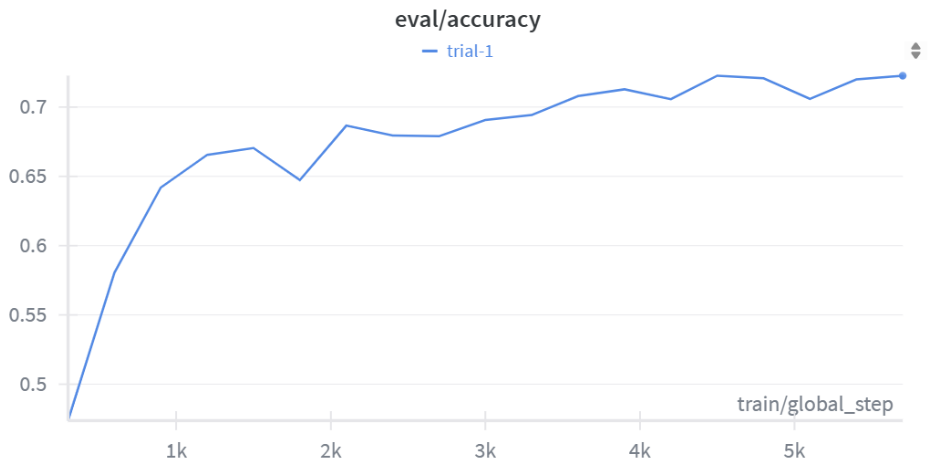


Figure 12: Accuracy for Electra using HF Code.

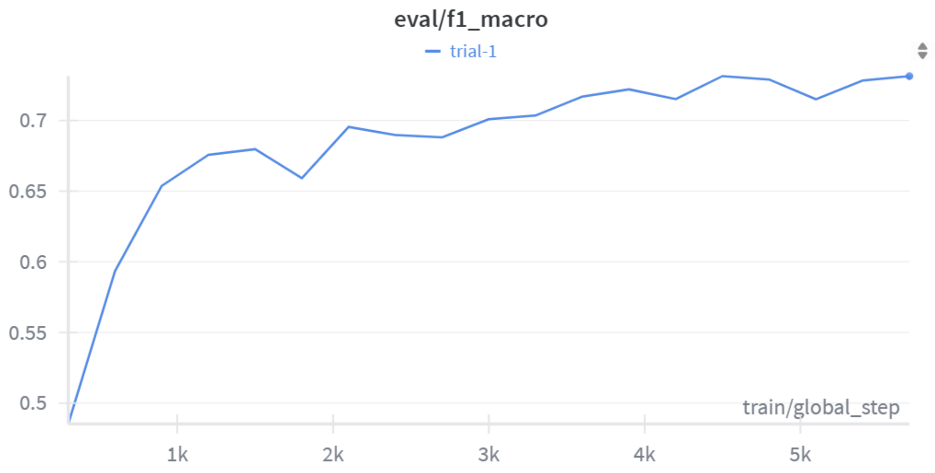


Figure 13: F1 for Electra using HF Code.