Tweet Sentiment Analysis Using Deep NLP Models and Compression Methods

Iliya Morgunov, Eadan Schechter*

August 21, 2025

Abstract

This paper explores the use of transformer-based models, specifically BERTweet and DeBERTa, for tweet sentiment classification. We trained and evaluated the models' performances based on a Kaggle dataset which includes above 40,000 tweets. Through evaluation, BERTweet and DeBERTa achieved comparable performance, with DeBERTa slightly outperforming on some metrics. To address the computational demands of these models, we applied compression techniques including quantization, pruning, and knowledge distillation, effectively reducing model size while maintaining high performance. Our findings highlight the potential of BERT-based models in tweet sentiment analysis and the importance of model optimization for deployment in resource constrained environments. Interestingly, distilled student models surpassed their larger teacher counterparts in accuracy, showing that compression can enhance performance beyond efficiency gains. Future work may explore integrating these models into META systems for automated tweet screening and sentiment-based filtering.

1 Introduction

The rapid proliferation of social media platforms, particularly Twitter, has fundamentally transformed the way individuals and organizations communicate, express opinions, and respond to global events. During the COVID-19 pandemic, Twitter became a crucial medium for disseminating news, monitoring public sentiment, and sharing real-time reactions to unfolding developments [1]. With millions of tweets generated daily, sentiment analysis of this content offers valuable insights for public health agencies, policymakers, and researchers seeking to understand how people felt about COVID-19, social dynamics, detect misinformation, all in order to make informed decisions.

However, sentiment analysis of tweets presents significant challenges. Tweets have characteristics that are generally different from those of traditional written text. This difference is seen with typically short length, use of informal grammar and often contain hashtags, mentions, abbreviations, URLs, and rapidly evolving slang [2].Thus, there is a challenge in applying existing language models pretrained on formal large-scale text to analyze such tweets, as they will struggle to capture the nuanced meanings and contextual cues inherited in them.

Recent advances in Natural Language Processing (NLP), particularly the development of transformer-based models, have revolutionized the field of sentiment analysis. Architectures such as BERT [3], RoBERTa [4], and DeBERTa [5] have achieved state-of-the-art results by leveraging deep contextual understanding and large-scale pre-training. Domain-specific models like BERTweet [2], specifically pre-trained on Twitter data, have further improved performance on tweet sentiment tasks. Unlike general-purpose models such as DeBERTa, which are trained primarily on formal text, BERTweet's exposure to millions of tweets enables it to better capture the slang, abbreviations, and noisy structure typical of Twitter data.

Despite these advances, deploying transformer models in real-world applications remains challenging due to their substantial computational and memory requirements. This limits their use in resource-constrained environments, such as mobile devices. To address this, model compression techniques - including quantization, pruning, and knowledge distillation have emerged as effective solutions to reduce the size and computational requirements of NLP models without significantly compromising their performance and inference time [6, 7].

In this work, we evaluate BERTweet and DeBERTa, for sentiment classification on a large corpus of COVID-19-related tweets. We systematically evaluate their performance via Pytorch and Hugging Face (HF) fine-tuning processes, and explore the application of model compression techniques to enable efficient, scalable deployment. We further introduce structured preprocessing using explicit tokens (`[url]`, `[date]`, `[location]`) to ensure metadata can be utilized effectively during tokenization and model training. Our findings highlight the effectiveness of deep NLP models for nuanced sentiment analysis, and emphasize the importance of model optimization for practical use in diverse computational environments.

[*] Code available at: GitHub Repository

2 Related Work

The task of sentiment classification on social media, and particularly Twitter, has attracted considerable research attention over the years. Early approaches relied on classical machine learning methods such as logistic regression, Random Forest, SVM, as well as deep learning methods of CNN and LSTM networks [8]. While effective to a degree, these methods struggled to capture the informal and context-dependent nature of Twitter language.

The introduction of deep learning, especially transformer-based models, has revolutionized the field of NLP. Models such as BERT [3], RoBERTa [4], and DeBERTa [5] have demonstrated outstanding performance on a range of text classification tasks, including sentiment analysis. Notably, domain-adapted transformers like BERTweet [2], pre-trained on large-scale Twitter data, have been shown to outperform general-purpose models on social media datasets. These findings indicate that transformer models, particularly BERT and its variants, offer significant advantages in terms of accuracy and robustness. However, general-purpose transformers often struggle with the noisy nature of Twitter data, where slang, abbreviations, hashtags, and nonstandard grammar are pervasive. These characteristics highlight the importance of domain adaptation and explicit metadata encoding.

Alongside advances in model architecture, the importance of robust data pre-processing and feature engineering remains central to effective sentiment analysis on Twitter. Prior research has demonstrated that incorporating tweet-specific features such as hashtags and mentions significantly improves classification performance [9]. Recent studies further suggest that hybrid approaches, which combine traditional pre-processing and feature engineering together with deep contextual embeddings from transformer-based models, lead to substantial gains in sentiment classification accuracy [2]. This combination between pre-processing, tweet-specific features, and powerful language models underpins the state-of-the-art in tweet sentiment analysis. Building on this, our work explicitly encodes tweet metadata into the text stream via structured tokens, allowing transformer tokenizers to natively process these cues.

Despite the strong performance of large transformer models, their deployment in practical applications is often limited by significant computational constraints. To address these challenges, researchers have increasingly explored model compression techniques such as quantization, pruning, and knowledge distillation to reduce model size and inference latency while preserving accuracy [6, 7]. These strategies have proven essential for scaling sentiment analysis to real-time applications and enabling efficient deployment on resource-constrained devices. Xu and McAuley [7] provide a comprehensive review, highlighting how these compression methods can substantially reduce the computational footprint of transformer-based models without significant loss in classification performance.

This study builds on these developments by evaluating both generic and Twitter-specific transformer models (DeBERTa, BERTweet) for tweet sentiment classification, and by systematically exploring model compression techniques to enable efficient deployment.

3 Exploratory Data Analysis

3.1 The Dataset

The dataset used in this study is the *CoronaNLP* dataset, sourced from Kaggle [1]. It consists of over 40,000 English-language tweets related to the COVID-19 pandemic, spanning March and April 2020. Each record contains the tweet text, timestamp, for most records a location, and a sentiment label annotated as one of five classes: `Extremely Negative`, `Negative`, `Neutral`, `Positive`, or `Extremely Positive`. The availability of timestamp and location fields in addition to the tweet text enables a richer, multidimensional analysis of sentiment trends over time and geography. Only the `OriginalTweet`, `Sentiment`, `TweetAt` (date), and `Location` columns were utilized for training the models for sentiment classification and exploratory analysis. Tweets which had no alphanumeric content were removed, thus the cleaned dataset contained 41,152 tweets.

3.2 Exploration of the Data

We conducted a comprehensive exploratory data analysis to understand the unique characteristics and challenges of the COVID-19 tweet sentiment dataset. The training dataset initially had **N = 41,157**

English-language tweets spanning March and April 2020, with 5 tweets lacking meaningful content, which were removed.

Initial inspection confirmed that the only missing values were in the `Location` column, with no duplicate tweets present. Tweets in the dataset showed substantial variability in length, language style, and content. The average tweet length was approximately **31 words**, with a maximum of **64 words** (Figure 1). Neutral tweets on average were shorter, whereas tweets expressing extreme sentiment tended to be longer. This variability in text properties highlights the informal and dynamic nature of social media communication. Frequent n-gram analysis further

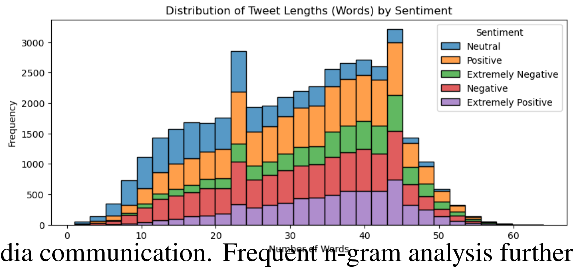
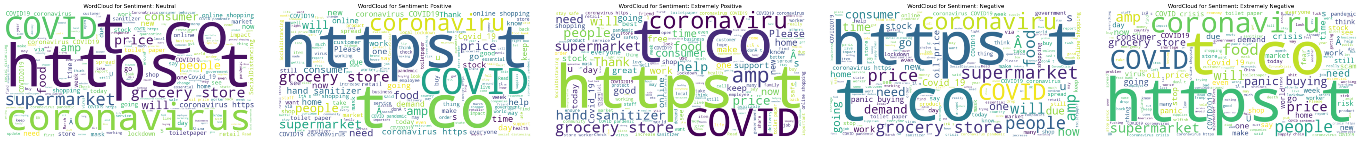


Figure 1: Distribution of Tweet Lengths (Words) by Sentiment

reveals that while the phrase "covid 19" dominates all classes, each sentiment class exhibits distinctive multi-word expressions that reflect public attitudes during the pandemic.

The dataset exhibits moderate label imbalance: **11,422 = 27.8%** tweets are labeled `Positive`, **9,917 = 24.1%** `Negative`, with the remainder divided among `Neutral` (**7,713 = 18.7%**), `Extremely Positive` (**6,624 = 16.1%**), and `Extremely Negative` (**5,481 = 13.3%**). The under representation of the extreme sentiment classes informed our use of stratified sampling and the use of macro-averaged metrics during model development. Specifically the maximization of the F1 metric (harmonic mean) during optimization.

Tweets frequently include features such as URLs, hashtags, and mentions. Specifically, approximately **48%** include a URL and more than 55% across all sentiments contain at least one hashtag. The prevalence of these features varies by sentiment class. While `Neutral` tweets include the most URLs and hashtags by percentage, the `Extremely Negative` tweets contain the least amount by percentage. To visualize these trends, we present word clouds for each sentiment class (Figure 2), illustrating the dominance of URLs is each sentiments via "https", as well as other most frequent found words under each sentiment class.



(a) Neutral (b) Positive (c) Extremely Positive (d) Negative (e) Extremely Negative

Figure 2: Word Clouds for Each Sentiment

The temporal analysis revealed that sentiment trends varied across days of the week and between the months of March and April. Geographically, tweets were sourced from a broad range of locations, with 8,593 tweets had unknown or missing location fields, meaning this information was not available for all tweets.

We checked additional tweet text features per sentiment to enrich the analysis: these include the ratio of ALL CAPS words, frequency of special punctuation, occurrence of COVID-19 related keywords, and presence of numbers. While none of these features alone strongly separated sentiment classes, they provided nuanced insight into how emotion, urgency, and topic salience were expressed in tweets.

To ensure robust model training, we thoroughly checked that the tweets have meaningful content. Only tweets containing no alphanumeric content were excluded, retaining only information-rich language suitable for NLP modeling. After cleaning, the final dataset comprised **41,152** tweets.

In addition to standard textual analysis, we leveraged the rich multivariate metadata present in the dataset. We designed a structured pre-processing pipeline to encode the metadata of dates, locations, and tweet features such as URLs, hashtags and mentions. By clearly differentiating with custom segment tokens between the tweet features and metadata, we enable downstream models to leverage both textual and contextual information.

Overall, our exploratory analysis revealed both the richness and complexity of the dataset. The findings directly informed our pre-processing strategy and modeling choices, emphasizing the necessity of tailored feature engineering and cleaning techniques for effective sentiment analysis on the received Twitter data.

4 Pre-processing and Feature Engineering

To maximize the effectiveness of transformer-based models for sentiment classification, we designed a structured, metadata-aware pre-processing pipeline. The raw tweets were first removed if they had non alphanumeric content. For each tweet, we extracted and encoded the structural features of hashtags, mentions, URLs, date, and location using explicit segment tokens (e.g., `[hashtags]`, `[mentions]`, `[url]`, `[date]`, `[location]`) added before each feature. The original symbols of hashtags(#) and mentions(@) were removed to enhance consistency and minimize tokenization ambiguity. URLs were resolved to descriptive titles where possible, and each URL was marked by a separate `[url]` segment to avoid confusion. Tweets with unreachable or uninformative URLs were excluded from further processing. As part of this step, we implemented a lightweight web scraping procedure to resolve shortened or raw links into human-readable page titles. This enrichment ensured that tokenizers processed semantically meaningful text rather than fragmented tokens such as "https" or random alphanumeric strings. To guarantee reproducibility, all resolved results were cached locally, avoiding repeated network calls. Links that returned errors, redirects, or non-informative placeholders were excluded to reduce noise. This scraping-based enrichment made the `[url]` segments both compact and informative, directly improving tokenizer compatibility and downstream model performance. The remaining tweet text that was not part of these features was appended after an initial `[tweet]` token.

Date and Location were added in a concise format (e.g., "March 16, 2020", "Vagabonds"), balancing clarity and token efficiency. This incremental, three-stage appending approach-progressively adding URLs, then date, then location - ensured each enriched tweet contained only meaningful, non-redundant metadata. All preprocessing decisions were guided by two goals in mind - clarity and minimal input length for each model, ensuring that all tweets remained well within the models token limit. The resulting representation enables models to leverage both the tweet content and contextual data crucial for robust sentiment analysis. This pre-processed tweet was then tokenized using the built-in tokenizers provided for each model from the Hugging Face platform.

Location	**TweetAt**	**OriginalTweet**	**TweetWithDateLocation**
Vagabonds	3/16/2020	Coronavirus Australia: Woolworths to give elderly, disabled dedicated shopping hours amid COVID-19 outbreak `https://t.co/bInCAVy8pP`	[tweet] Coronavirus Australia: Woolworths to give elderly, disabled dedicated shopping hours amid COVID-19 outbreak [url] Coronavirus Australia: Woolworths to give elderly, disabled dedicated shopping hours amid COVID-19 outbreak [date] March 16, 2020 [location] Vagabonds

Table 1: Tweet Transfomation

5 Model Selection

For sentiment classification, we selected two BERT-based transformer models - BERTweet and DeBERTa - chosen for their proven effectiveness on social media and sentiment tasks.

BERTweet [2] is a RoBERTa-based model specifically pre-trained on 850 million English tweets. Its architecture and training corpus make it particularly well-suited for capturing the unique and informal language, slang, and multimodal features prevalent on Twitter, including mentions, hashtags, and URLs. The BERTweet corpus includes tweets from both the general Twitter stream and a COVID-19–specific collection (January–March 2020), a slight overlap with our data period. As a result, BERTweet is particularly well-equipped to handle the vocabulary, topic, and sentiment nuances present in our dataset. Previous research has shown that BERTweet achieves state-of-the-art (SOTA) results for tweet sentiment analysis, outperforming general domain models by a significant margin.

In addition, we evaluated DeBERTa (Decoding-enhanced BERT with Disentangled Attention) [5], an advanced transformer model developed by Microsoft that introduces disentangled attention mechanisms, explicitly separating content and positional embeddings, and an enhanced decoding scheme. DeBERTa has demonstrated SOTA results across a range of NLP benchmarks, including sentiment analysis on both traditional and social media datasets. Recent studies have further highlighted DeBERTa's effectiveness for Twitter sentiment analysis: for example, Assiri et al.[10] combined DeBERTa with a recurrent unit (GRU) and reported an F1 score of approximately 97% on a large Twitter sentiment dataset, illustrating DeBERTa's ability to capture nuanced sentiment cues in short, informal texts. These results underscore DeBERTa's robustness and adaptability for sentiment classification in the social media domain.

Both models were fine-tuned on our preprocessed dataset using the HF library and PyTorch. For each model, we performed 3-fold stratified cross-validation and conducted hyperparameter tuning using the Optuna framework, optimizing for macro-averaged F1 score. The main hyperparameters explored included learning rate, batch size, and weight decay. Training was performed for up to 25 epochs with early stopping that was employed based on validation F1 to prevent overfitting. The training process was conducted on a GPU environment via Google Colab (NVIDIA A100), and all experiments were tracked using the Weights & Biases (W&B) platform to monitor the different training and validation metrics. To match sequence length with the representation used, we adjusted the maximum sequence length during training. In initial experiments on the *raw* tweets we set `max_len`=64 to avoid unnecessary padding, consistent with the short tweet lengths observed in EDA. After introducing the structured tokens (`[url]`, `[date]`, `[location]`) and assembling the *enriched* representation, we increased the limit to `max_len`=128 to ensure the added metadata segments were not truncated. This change preserved full coverage of the tweet and metadata while keeping batches efficient (dynamic padding) and GPU memory usage stable; empirically, truncation was negligible under the 128 - token cap. Final models were retrained on the complete training set using the best hyperparameters and evaluated on the test set.

By leveraging both domain-adapted and general-purpose transformer architectures, our experimental design enables a rigorous evaluation of the impact of model pretraining on sentiment analysis performance in the context of COVID-19 Twitter data. In later stages, these fully fine-tuned models also served as teachers for knowledge distillation into smaller student variants, enabling us to compare not only raw model performance but also the effectiveness of transferring knowledge into compact architectures.

5.1 Baselines

To anchor performance and probe the benefit of explicit structure, we implemented three baselines. Our motivation was to test how much domain adaptation and structured preprocessing matter, even when models are not fine-tuned.

First, a *zero-shot* classifier using `facebook/bart-large-mnli` with the hypothesis template "This text expresses {} sentiment." and candidate descriptors mapped directly to the five sentiment labels, where predictions are taken from the argmax of returned probabilities. This formulation follows the textual entailment approach to zero-shot classification proposed by Yin et al. [13], where labels are expressed as natural language hypotheses.

Second, a *few-shot* cosine–prototype classifier using `sentence-transformers/all-MiniLM-L6-v2`, in which tweets are embedded (mean pooled and L2-normalized), prototypes are computed as the average of k=100 labeled exemplars per class, and test tweets are assigned to the nearest prototype. This simple nearest-prototype strategy is grounded in the framework of Prototypical Networks [14], which demonstrate that averaging exemplars yields robust and data-efficient class representations.

Third, we applied a *supervised baseline* with `vinai/bertweet-base` fine-tuned directly on `OriginalTweet` (*Raw*), serving as a strong reference point before adding metadata or applying a compression technique.

To test whether simple methods also benefit from structured cues, the zero-shot and few-shot baselines were run on both *Raw* and the enriched representation concatenating `[url]`, `[hashtags]`, `[date]`, and `[location]`. Although these baselines remained relatively weak overall, the enriched variants consistently outperformed their raw counterparts, showing that preprocessing helps even without fine-tuning.

6 Model Compression Techniques

Following the fine-tuning and evaluation of our two selected models, BERTweet and DeBERTa, we applied three compression techniques to reduce model size and improve inference speed to enable deployment on resource-constrained devices. First, we applied post-training quantization using PyTorch's dynamic quantization. Specifically, we quantized all linear layers, converting their weights and activations from 32-bit floating point to 8-bit integers during inference. This reduces model size and memory footprint by lowering parameter precision, enabling faster inference. However, in our experiments quantization severely degraded accuracy, showing that this approach was unsuitable for sentiment classification in our setting. Second, we applied global unstructured pruning, zeroing out the lowest-magnitude weights across all linear layers until reaching 30% sparsity. This produced a sparser, lighter model with reduced memory footprint and modest accuracy degradation. Lastly, we applied knowledge distillation by training a smaller, more compact student model (*BERT-tiny* [11] for BERTweet, and *DeBERTa-v3-small* [12] for DeBERTa) to mimic the outputs of the fully fine-tuned teacher models. Despite the "small" name, v3-small uses a 128k subword vocabulary; its token embedding matrix (128k×768) dominates the checkpoint size. Consequently, the distilled student achieves lower inference latency (fewer layers) but similar or larger parameter count/model size than DeBERTa-base. This KD therefore compresses compute, not necessarily disk size. During distillation, the student was trained to minimize a combined loss: the hard target cross-entropy loss with ground-truth labels, and the soft target Kullback-Leibler divergence loss, matching the probability distributions (soft targets) output by the teacher, with temperature scaling applied. We applied this training using the HF API, keeping the teacher model frozen. The student model HP were selected using 3-fold stratified CV on the training set and optimized with Optuna to maximize macro-F1 averaged across folds, with all HP search experiments logged using W&B. Contrary to expectation, our distilled student models did not merely approximate teacher performance; they consistently surpassed their larger teachers in macro-F1, highlighting that distillation acted as an effective form of regularization in addition to compression.

7 Results

Model	**Loss(*CE*)**	**F1**	**Accuracy**	**Precision**	**Recall**	**Inference Time ($\frac{sec}{sample}$)**	**Model Size (*Mb*)**	**Parameter Count**
Baseline - BART-large-MNLI - Zero-shot (raw)	2.4844	0.1381	0.2954	0.5494	0.2196	0.1026	-	-
Baseline - BART-large-MNLI - Zero-shot	2.5196	0.1527	0.3041	0.3870	0.2280	0.1020	-	-
Baseline - all-MiniLM-L6-v2 - Few-shot (raw)	1.6000	0.3131	0.3060	0.3081	0.3246	0.0003	-	-
Baseline - all-MiniLM-L6-v2 - Few-shot	1.6012	0.3259	0.3210	0.3224	0.3464	0.0003	-	-
Baseline - BERTweet (raw)	0.9788	0.6219	0.6058	0.6258	0.6243	0.0014	539.6995	134903813
BERTweet - PyTorch	1.0509	0.6111	0.5964	0.6174	0.6069	0.0020	539.6985	134903813
BERTweet - HF	1.1460	0.6261	0.6111	0.6275	0.6248	0.0015	539.6404	134903813
BERTweet - PyTorch Quantized	1.5785	0.1664	0.3136	0.6163	0.2388	0.0293	283.1712	49291776
BERTweet - PyTorch Pruned	1.0499	0.5791	0.5714	0.5973	0.5818	0.0023	539.6995	134903813
BERTweet - PyTorch Distilled	0.7142	0.7499	0.7362	0.7549	0.7483	0.0003	**17.5631**	**4386565**
BERTweet - HF Quantized	2.0628	0.2179	0.2809	0.3152	0.2909	0.0215	283.1712	49291776
BERTweet - HF Pruned	1.1164	0.5923	0.5908	0.6337	0.5933	0.0022	539.6995	134903813
BERTweet - HF Distilled	0.7224	0.7477	0.7354	0.7509	0.7470	**0.0002**	**17.5631**	**4386565**
DeBERTa - PyTorch	0.9614	0.6207	0.6051	0.6205	0.6250	0.0026	556.8687	139196165
DeBERTa - HF	0.9803	0.6421	0.6280	0.6458	0.6393	0.0019	556.8098	139196165
DeBERTa - PyTorch Quantized	1.5297	0.2514	0.3133	0.4802	0.2913	0.0461	257.8754	39446784
DeBERTa - PyTorch Pruned	1.0263	0.5726	0.5569	0.5803	0.5880	0.0024	556.8697	139196165
DeBERTa - PyTorch Distilled	0.4541	**0.8577**	**0.8509**	**0.8607**	0.8572	0.0010	567.6399	141898757
DeBERTa - HF Quantized	1.9091	0.3920	0.4018	0.4556	0.4165	0.0481	257.8754	39446784
DeBERTa - HF Pruned	0.9785	0.6109	0.5961	0.6140	0.6195	0.0023	556.8697	139196165
DeBERTa - HF Distilled	**0.4172**	0.8539	0.8468	0.8542	**0.8575**	0.0005	567.6399	141898757

Table 2: Comparison of model performance and baseline (both raw and transformed data) metrics for all model variants and methods. All scores are macro-averaged across the five sentiment classes.

Overall, the results reveal clear trends rather than isolated numbers. The zero-shot and few-shot baselines, even when enriched with metadata tokens, remained weak (F1 ≈ 0.13–0.33), confirming the difficulty of this task without domain adaptation. In contrast, fully fine-tuned BERTweet and DeBERTa achieved strong performance with nearly identical F1 scores (BERTweet 0.61–0.63; DeBERTa 0.62–0.64), establishing solid teacher models. This suggests that domain pretraining and advanced architecture offered similar benefits on this dataset. Among compression methods, quantization consistently degraded performance and made inference slower, while pruning caused only moderate performance drops with little change in efficiency. Distillation stood out as the students performance surpassed their teachers, underscoring distillation as both an effective compression and regularization strategy.

7.1 Comparison between BERTweet and DeBERTa

BERTweet benefitted from its domain-specific pretraining on large-scale Twitter data, which gave it an advantage in handling slang, hashtags, and informal grammar. However, our experiments showed that DeBERTa, especially when distilled, achieved the strongest overall results, though the base models were nearly tied. This suggests that while domain adaptation provides important gains, advanced architectures with effective compression can surpass even well-tailored domain-specific models. The training curves of both models, shown in Figure 3, further illustrate that their performance metrics are closely aligned, with DeBERTa showing a slight edge.

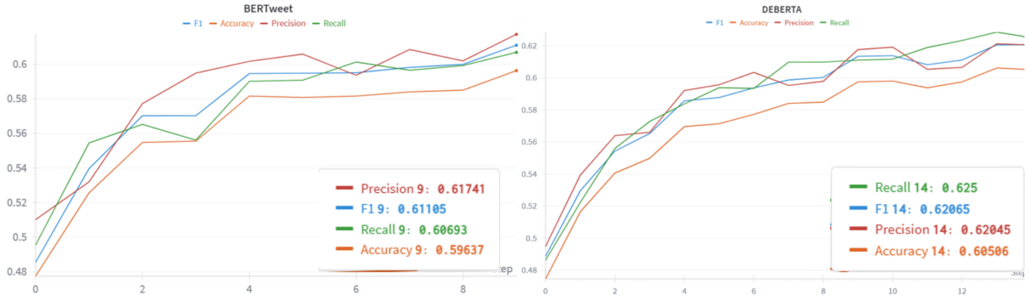


Figure 3: Training curves comparing BERTweet and DeBERTa across F1, accuracy, precision, and recall on evaluation set. The results indicate that while both models perform similarly, DeBERTa shows a slight advantage, particularly when distilled.

7.2 Comparison of Compressed Models

A notable and somewhat unexpected finding is that distilled models surpassed their respective teacher models in performance. Distilled DeBERTa achieved an F1 score of 0.85 compared to 0.64 for the teacher, though without reducing model size. Distilled BERTweet, in contrast. improved to 0.75 from 0.62 with a much smaller model size (17 MB vs. 540 MB) . This suggests that distillation may act as a form of regularization, smoothing over noise in the training data and leading to better generalization.

Our compression methods demonstrate contrasting trade-offs: quantization reduced model size but at the cost of degraded performance, pruning caused moderate performance drops with slight changes in efficiency, while distillation yielded student models that surpassed teacher performance. This highlights that model compressions can simultaneously improve efficiency and with distillation, even enhance performance, enabling deployment of high-performing sentiment classifiers in constrained environments.

8 Summary

This study demonstrates two central findings. First, structured preprocessing with explicit tokens (`[url]`, `[date]`, `[location]`) proved highly effective: it preserved contextual cues in a tokenizer-friendly format and delivered measurable gains even for weak zero-shot and few-shot baselines. Second, knowledge distillation did not merely preserve teacher performance but consistently surpassed it, showing that compression can also act as a form of regularization and yield better performing models. Together, these results highlight that careful preprocessing and distillation are complementary strategies-improving both efficiency and accuracy in transformer-based sentiment analysis of tweets.

At the same time, the base models results indicate that there is room for improvement. Additional feature engineering or richer metadata could have potentially led to better performances. Our token-based transformation, while systematic, may have imposed unfamiliar symbols that the models had not been exposed to during pretraining; effectively learning these tokens might have required larger-scale data or training from scratch, as is done with special tokens such as `[CLS]` or `[MASK]`. Furthermore, although we experimented with data augmentation, the approach was not fully pursued due to project scope. Future work could revisit augmentation via paraphrasing (e.g., pre-trained models like PEGASUS or T5) or back-translation (EN→SE→EN), combined with heuristic filters such as cosine similarity to retain only high-quality variants. Such augmentation should be applied only to the `[tweet]` content, since metadata tokens like `[url]` and `[location]` are fixed identifiers and cannot be safely paraphrased.

References

[1] ”Coronavirus tweets NLP - Text Classification”, Kaggle, 2020. https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification

[2] D. Q. Nguyen, T. Vu, and A. T. Nguyen, ”BERTweet: A pre-trained language model for English Tweets”, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 9–14, 2020.

[3] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, ”BERT: Pre-training of deep bidirectional transformers for language understanding”, arXiv preprint arXiv:1810.04805, 2018.

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, ”RoBERTa: A robustly optimized BERT pretraining approach”, arXiv preprint arXiv:1907.11692, 2019.

[5] P. He, X. Liu, J. Gao, and W. Chen, ”DeBERTa: Decoding-enhanced BERT with disentangled attention”, arXiv preprint arXiv:2006.03654, 2021.

[6] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, ”DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”, arXiv preprint arXiv:1910.01108, 2019.

[7] C. Xu and J. McAuley, ”A survey on model compression for natural language processing”, arXiv preprint arXiv:2202.07105, 2022.

[8] S. Rosenthal, N. Farra, and P. Nakov, ”SemEval-2017 Task 4: Sentiment Analysis in Twitter”, In ”Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)”, pp. 502–518, 2017.

[9] S. M. Mohammad, S. Kiritchenko, and X. Zhu, ”Twitter sentiment analysis: The good, the bad and the OMG!”, In Proceedings of the International AAAI Conference on Web and Social Media, vol. 7, no. 1, pp. 538–541, 2013.

[10] A. Assiri, A. Gumaei, F. Mehmood, T. Abbas and S. Ullah, ”DeBERTa-GRU: Sentiment Analysis for Large Language Model”, Computers, Materials & Continua, vol. 79, no. 3, 2024.

[11] Turc et al., ”Well-Read Students Learn Better: On the Importance of Pre-training Compact Models”, arXiv preprint arXiv:1908.08962, 2019.

[12] P. He, J. Gao and W. Chen, ”DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing”, arXiv preprint arXiv:2111.09543, 2021.

[13] W. Yin, J. Hay, and D. Roth, ”Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach”, arXiv preprint arXiv:1909.00161, 2019.

[14] J. Snell, K. Swersky, and R. Zemel, ”Prototypical Networks for Few-shot Learning”, arXiv preprint arXiv:1703.05175, 2017.