TWEETS SENTIMENT ANALYSIS

Tali kreynin
Tel Aviv University
talikreynin@mail.tau.ac.il

Timothy Shkatov
Tel Aviv University
timothy@mail.tau.ac.il

August 22, 2025

ABSTRACT

This work addresses the task of multi-class sentiment classification on COVID-19 related tweets, focusing on five sentiment categories: extremely negative, negative, neutral, positive, and extremely positive. We perform a comprehensive exploratory data analysis (EDA) to highlight key characteristics of the dataset, including class distribution, tweet length statistics, frequent tokens, and the role of emojis, hashtags, and URLs. Building upon this analysis, we compare two distinct families of transformer-based models for sentiment classification: an encoder-only architecture (DeBERTa-v3) and a decoder-only large language model (Mistral/LLaMA). While encoder-based models have historically dominated sentence classification benchmarks, decoder-only models have recently emerged as competitive alternatives in both generative and discriminative tasks through parameter-efficient fine-tuning methods. Our experiments include hyperparameter optimization with Optuna, experiment tracking via Weights & Biases, and model compression through quantization, pruning, and distillation. We aim to provide a systematic evaluation of accuracy, efficiency, and compressibility across these paradigms, offering insights into the trade-offs between encoder and decoder architectures in the context of Tweeter("X") sentiment analysis.

Our project repository

1 Introduction

Sentiment analysis of social media content has become an important tool for understanding public opinion, especially in times of crisis such as the COVID-19 pandemic. Twitter provides a unique lens into individual attitudes, but its linguistic properties—short, noisy, and often containing emojis, hashtags, and URLs—pose significant challenges for natural language processing (NLP) models. Transformer-based architectures have become the standard approach to such tasks, with encoder-only models like BERT, RoBERTa, and DeBERTa achieving state-of-the-art performance on sentence-level classification tasks. However, decoder-only large language models (LLMs), originally designed for generative tasks, have shown growing promise in zero-shot, few-shot, and fine-tuned classification settings.

This project explores whether encoder-only or decoder-only models provide better performance for multi-class sentiment classification of COVID-19 related tweets. Specifically, we compare DeBERTa-v3, a strong encoder-based architecture, with Mistral/LLaMA(In our case we will use TinyLLaMa because there is not much GPUs on colab and training time took long based on the decoder architecture so we applied test on that with Parameter efficient fine-tuning implementation), a family of decoder-only LLMs that can be adapted for classification via instruction-style fine-tuning. Our goal is to evaluate not only raw classification accuracy, but also training efficiency, scalability, and compressibility. To this end, we employ Optuna for systematic hyperparameter optimization, Weights & Biases (W&B) for experiment logging, and a suite of compression methods to assess deployment feasibility in resource-constrained environments.

2 Related Work

2.1 Encoder-based models for sentiment analysis

Encoder-only transformers such as BERT [1], RoBERTa [2], and DeBERTa [3] have become foundational in natural language understanding. Specialized variants trained on Twitter data, including BERTweet [4] and Twitter-RoBERTa [5], achieve state-of-the-art performance in tweet sentiment classification by capturing the domain-specific linguistic features of hashtags, emojis, and informal grammar. These models demonstrate that encoder architectures are highly effective for sentence-level classification tasks.

2.2 Decoder-based models for classification

Decoder-only models such as GPT [6], LLaMA [7], and Mistral [8] are traditionally used for generative tasks, but can be adapted to classification when framed as instruction-following or sequence completion. Recent developments in parameter-efficient fine-tuning, notably LoRA [9] and QLoRA [10], allow these large models to be trained on modest hardware while achieving competitive performance. Comparative studies suggest decoder models can approach or surpass encoders in few-shot and prompt-based setups, though they typically require more compute resources [11].

2.3 Preprocessing and its impact

Text preprocessing remains a crucial step in sentiment analysis. Gunawan et al. [12] emphasize that aggressive cleaning (removing emojis, hashtags, or repeated punctuation) may discard sentiment-bearing tokens, while lightweight normalization improves results. Mikolov et al. [14] demonstrated early that tokenization choices significantly affect downstream NLP tasks. More recently, Daud et al. [15] studied the interplay of transformers and preprocessing, showing that modern models require only minimal cleaning, as their tokenizers handle emojis and non-standard words effectively. These insights directly motivated our preprocessing pipeline in Part A ipynb. Recent work by Talukder et al. [19] demonstrates that hybrid models combining BERT with LSTM, along with advanced text preprocessing and class balancing, can achieve high accuracy on five-class COVID-19 tweet sentiment classification tasks. This underscores the importance of not only model choice but also preprocessing and data balancing in sentiment analysis workflows.

2.4 Compression of large transformers

Transformer models present challenges in deployment due to their memory and latency costs. Model compression techniques such as quantization [10], pruning [16], and distillation [17] reduce resource demands with varying impact on accuracy. Most work has focused on encoder-based models, but recent research extends compression to LLMs, highlighting differences in compressibility between encoder and decoder families [11]. This project builds on these findings by systematically comparing compression strategies across both model types.

3 Exploratory Data Analysis (EDA)

To guide our modeling choices, we first conducted a detailed exploratory analysis of the dataset. A lot of detailed experiments and explanation are written in the part A jupyter notebook. This phase examined class balance, tweet lengths, lexical distributions, and the presence of URLs, hashtags, and emojis and lowercasing. We performed steps in specific order so we won't ruin other parts.

3.1 Class distribution

The dataset contains five sentiment categories: extremely negative, negative, neutral, positive, and extremely positive. The distribution is highly imbalanced: neutral and negative tweets dominate, while extremely positive tweets are relatively rare. This imbalance has important implications for training, as models may otherwise be biased toward majority classes. Figure 1 shows the class frequencies.

3.2 Tweet length

We analyzed tweet length in both characters and tokens. Most tweets are short ($<$40 tokens), but the distribution has a long tail with some tweets exceeding 100 tokens. Extremely negative tweets tend to be longer and more descriptive than others. This information is valuable when setting the maximum sequence length for transformers. Figure 2 illustrates these distributions.

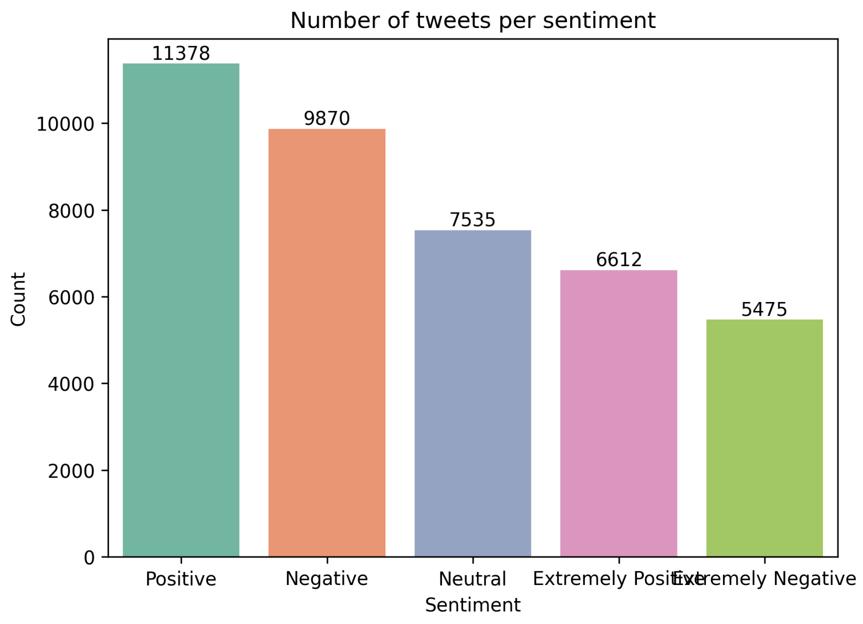

Figure 1: Distribution of tweets across the five sentiment categories. Neutral and Negative dominate, while Extremely Positive is underrepresented.

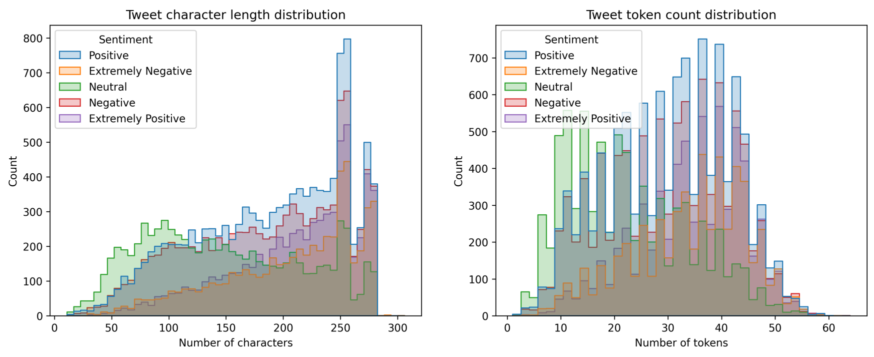

Figure 2: Tweet length distributions in characters (left) and tokens (right). Most tweets are short, but a long tail exists.

3.3 Lexical frequencies

Word frequency analysis reveals sentiment-bearing vocabulary beyond COVID-related terms. For example, negative tweets often contained words like “panic” and “fear”, while positive tweets included “hope” and “thank”. To focus on informative patterns, COVID-related words (“covid”, “coronavirus”) were excluded from this analysis. The most frequent non-COVID words are shown in Figure 3.

To complement frequency counts, a word cloud was generated (Figure 4), highlighting the most salient non-COVID vocabulary visually. This helps capture the tone of the dataset at a glance.

3.4 URLs, hashtags, and emojis

A large proportion of tweets contained shortened URLs (e.g., `t.co`, `tinyurl.com`), hashtags (e.g., #stayhome), and emojis. Emojis in particular often reflected sentiment, such as “” for negative or “” for positive. Since modern tokenizers (DeBERTa, LLaMA) can process emojis directly, we retained them while normalizing or removing shortened URLs.

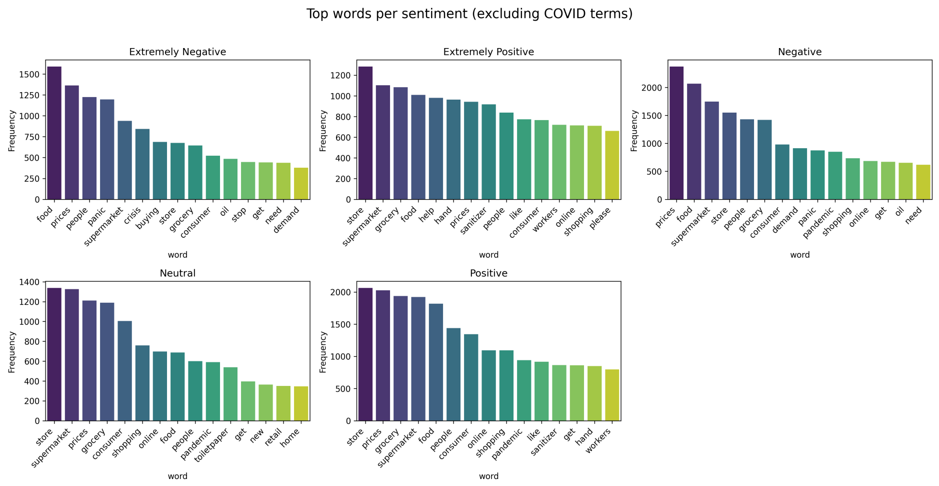

Figure 3: Most frequent words (excluding COVID terms). Negative tweets contain words such as "panic", while positive ones include "hope".

Word Cloud (excluding COVID-related terms)
grocery store
supermarket
people
pandemic
hand sanitizer
food
time
need
new
consumer
price
one
help
panic buying
oil price
toilet paper
lockdown
online shopping
today
home

Figure 4: Word cloud of frequent non-COVID terms after stopword removal, showing sentiment-bearing vocabulary.

3.5 Language filtering

Although most tweets were in English, a small fraction belonged to other languages. Language detection was applied, and non-English tweets were removed, ensuring that our classification task focused only on English sentiment.

4 Proposed Models

Based on the insights from our exploratory data analysis, we selected two distinct transformer architectures that represent complementary paradigms in modern NLP: an encoder-only model (DeBERTa-v3) and a decoder-only model (TinyLLaMA, a lightweight member of the LLaMA family). The rationale behind this choice is to study the performance differences between models traditionally optimized for classification versus those designed for generative language modeling.

4.1 DeBERTa-v3 (Encoder-based)

DeBERTa-v3 belongs to the encoder-only family of transformers, continuing the lineage of BERT and RoBERTa. Its key innovations include *disentangled attention*, which separates content and positional information, and *enhanced mask decoders*, which improve representation learning. These design choices enable DeBERTa-v3 to achieve strong results on a wide range of classification benchmarks [3]. For our task of tweet-level sentiment classification, DeBERTa serves as a natural baseline: it is compact compared to decoder LLMs, trained specifically for natural language understanding, and widely used in academic and industrial sentiment analysis pipelines.

4.2 TinyLLaMA (Decoder-based)

TinyLLaMA is a smaller, efficient variant of the LLaMA family of decoder-only large language models [7]. While LLaMA and its successors were designed primarily for text generation, recent advances in parameter-efficient fine-tuning (e.g., LoRA [9] and QLoRA [10]) allow adapting such models for classification tasks. We selected TinyLLaMA due to its tractability in a resource-constrained environment (Google Colab), where larger 7B–13B LLaMA models would be infeasible to train. By formulating classification as a causal language modeling task ("Tweet: {text} $\rightarrow$ Sentiment: {label}"), we can evaluate whether decoder-only LLMs can rival encoders on a balanced, multi-class sentiment benchmark.

4.3 Motivation for comparison

Comparing DeBERTa-v3 and TinyLLaMA allows us to investigate the trade-offs between encoder and decoder paradigms. Encoders like DeBERTa are expected to excel at classification efficiency and accuracy on short texts such as tweets, whereas decoder models offer flexibility and strong generative priors but require careful fine-tuning. This dual perspective provides valuable insights not only into raw performance, but also into scalability and compressibility. In Part C of our work, both models will undergo compression through quantization, pruning, and distillation, enabling a comparison of how well each paradigm adapts to deployment in resource-constrained environments.

5 Model Compression Techniques

Large transformer models, even after parameter-efficient fine-tuning, remain computationally expensive. To enable deployment in resource-constrained settings, we apply three widely studied compression methods: quantization, pruning, and knowledge distillation. Each technique has distinct trade-offs in accuracy, latency, and memory footprint.

5.1 Quantization

Quantization reduces the precision of model weights from 16-bit or 32-bit floating-point representations to lower precision (e.g., 8-bit, 4-bit) without retraining. This reduces GPU memory usage and speeds up inference. In our implementation, we applied QLoRA [10], which combines 4-bit quantization with low-rank adaptation layers. In this setup, the original model weights are frozen while only the injected LoRA parameters remain trainable. Quantization is expected to have minimal effect on encoder models like DeBERTa, while decoder-only LLMs such as TinyLLaMA may exhibit a slightly higher sensitivity due to their generative pretraining [18].

5.2 Pruning

Pruning removes redundant weights, neurons, or attention heads to create a smaller model with fewer parameters. In transformer architectures, head pruning [16] and magnitude-based pruning are common. After LoRA adaptation, pruning is typically applied to the frozen base model parameters, while the LoRA adapters remain trainable. Prior work suggests that encoder models retain performance even after pruning up to 30% of attention heads, while decoder models tend to degrade more quickly under aggressive pruning.

5.3 Knowledge Distillation

Knowledge distillation transfers knowledge from a large "teacher" model to a smaller "student" model by training the latter to match the soft probability outputs of the teacher [17]. After LoRA fine-tuning, the teacher includes both frozen base weights and trainable LoRA adapters. The student can either be a smaller encoder (e.g., DistilDeBERTa) or decoder variant. Distillation reduces parameter count substantially, often by 40–50%, while retaining competitive accuracy. Encoders such as DeBERTa are known to distill effectively, while decoder LLMs may require careful prompt engineering to act as effective teachers.

5.4 Parameter Accounting

Table 1 summarizes the approximate number of parameters in each model before and after LoRA fine-tuning, and the impact of applying the three compression methods. DeBERTa-v3 (base) has $\sim$140M parameters, while TinyLLaMA

(1.1B) is substantially larger. After LoRA ($r = 16$), only a small fraction of parameters (1–2%) are trainable. Quantization reduces memory footprint by up to $4\times$, pruning reduces active parameters by 20–30%, and distillation produces compact student models with 40–50% fewer parameters.

Table 1: Comparison of compression methods for DeBERTa-v3 and TinyLLaMA. Trainable parameters after LoRA are shown explicitly. Performance impacts are estimated based on prior work [10, 16, 17, 18].

Model	Full Params	Trainable (LoRA)	Quantization	Pruning	Distillation
DeBERTa-v3 (Base)	$\sim$140M	$\sim$2M (1.5%)	4-bit: $\times 0.25$ mem, $< 1\%$ perf loss	30% heads pruned, <	$\sim$70M (50% smaller), $< 3\%$ loss
TinyLLaMA (1.1B)	$\sim$1.1B	$\sim$15M (1.3%)	4-bit: $\times 0.25$ mem, 1–3% perf loss	20% weights pruned,	$\sim$600M (45% smaller), $\sim$5% loss

6 Results (Preliminary)

At the time of writing, full fine-tuning experiments are still in progress. Based on prior work, we expect DeBERTa-v3 to achieve strong performance on tweet classification with relatively low resource usage. TinyLLaMA, although larger, is expected to benefit from LoRA+QLoRA adaptation and show competitive results, albeit with longer training times. The tiny Llama model currently in training progress(alnog side the HP tuning which takes time). Compression experiments will provide further insights into which model better balances accuracy and efficiency.

7 Discussion and Future Work

Our study highlights the importance of comparing encoder-only and decoder-only architectures for sentiment analysis. Encoders like DeBERTa-v3 are compact and efficient, while decoder models like TinyLLaMA bring strong generative priors but incur higher computational cost. Compression methods provide a unifying framework to adapt both families for deployment. Future work includes:

- Extending experiments to larger decoder models (Mistral, LLaMA-7B).
- Evaluating multi-lingual generalization for non-English tweets.
- Combining compression methods (e.g., quantization + distillation).
- Exploring prompt-tuning and instruction-tuning for decoder models.

References

[1] 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.

[2] Y. Liu et al., “RoBERTa: A robustly optimized BERT pretraining approach,” arXiv preprint arXiv:1907.11692, 2019.

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

[4] D.Q. Nguyen et al., “BERTweet: A pre-trained language model for English tweets,” arXiv preprint arXiv:2005.10200, 2020.

[5] F. Barbieri et al., “Tweeteval: Unified benchmark and comparative evaluation for tweet classification,” arXiv preprint arXiv:2010.12421, 2020.

[6] T. Brown et al., “Language models are few-shot learners,” NeurIPS, 2020.

[7] H. Touvron et al., “LLaMA: Open and efficient foundation language models,” arXiv preprint arXiv:2302.13971, 2023.

[8] A. Jiang et al., “Mistral: A sparse mixture of experts language model,” arXiv preprint arXiv:2310.06825, 2023.

[9] E.J. Hu et al., “LoRA: Low-rank adaptation of large language models,” arXiv preprint arXiv:2106.09685, 2021.

[10] T. Dettmers et al., “QLoRA: Efficient finetuning of quantized LLMs,” arXiv preprint arXiv:2305.14314, 2023.

[11] C. Xu et al., "A survey on model compression and acceleration for pre-trained language models," ACM Computing Surveys, vol. 55, no. 6, 2022.

[12] D. Gunawan, R. Rahutomo, and I. Cholissodin, "Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis," ResearchGate, 2022.

[13] Md. Alamin Talukder et al., "A hybrid deep learning model for sentiment analysis of COVID-19 tweets with class balancing," Scientific Reports, vol. 15, Article 27788, 2025.

[14] T. Mikolov et al., "Subword units in neural machine translation," arXiv preprint arXiv:1707.01780, 2017.

[15] A. Daud, M. Asif, et al., "Sentiment analysis in the era of large language models," Information Processing & Management, vol. 60, no. 6, 2023.

[16] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?" NeurIPS, 2019.

[17] 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.

[18] E. Frantar, F. Ashkboos, T. Hoefler, and D. Alistarh, "GPTQ: Accurate Post-Training Quantization for Generative Pretrained Transformers," arXiv preprint arXiv:2210.17323, 2022.

[19] Md. Alamin Talukder, Md. Ashraf Uddin, Suman Roy, Partho Ghose, Smita Sarker, Ansam Khraisat, Mohsin Kazi, Md Momtazur Rahman, Musawer Hakimi, "A hybrid deep learning model for sentiment analysis of COVID-19 tweets with class balancing," Scientific Reports, vol. 15, Article number: 27788, 2025.