2025. Volume 9, Number 2 *Pages 213-239*

UDC: 004.8:004.912
DOI: https://doi.org/10.30546/09090.2025.002.10.1045

EVALUATING THE USABILITY AND EFFICIENCY OF COMPACT LARGE LANGUAGE MODELS FOR SENTIMENT ANALYSIS TASKS

Aydin GASIMOV*

[1]*Azerbaijan State Oil and Industry University,*
Baku, Azerbaijan

ARTICLE INFO

Article history:
Received: 2025-11-01
Received in revised form:2025-11-05
Accepted:2025-12-10
Available online

Keywords:
"Sentiment Analysis"
"Large Language Models"
"Efficiency"
"Zero-Shot Learning"
"Compact LLMs"
2010 Mathematics Subject Classification: 68T50 (primary), 68T07, 68T09, 68T35

ABSTRACT

The rapid proliferation of large language models (LLMs) has enabled substantial advances in natural language processing, but their resource requirements create barriers to accessibility, cost-efficiency, and sustainable deployment. This paper presents a systematic benchmark of compact LLMs—ranging from 135M to 8B parameters— on eight diverse sentiment analysis datasets, including binary, fine-grained, domain-specific, social media, and aspect-based tasks. Models were evaluated in a zero-shot setting using normalized accuracy metrics to ensure comparability across datasets of varying difficulty, alongside measurements of latency and memory usage.

Our findings reveal that mid-sized models in the 3–4B parameter range, particularly Gemma 3 4B and Microsoft Phi-4 Mini, consistently outperform or rival larger 7–8B models while offering significantly lower latency and memory footprints. Sub-1B models, while largely ineffective in zero-shot conditions, retain potential for high-throughput pipelines and fine-tuned deployments. Conversely, larger 7–8B models remain valuable for accuracy-critical tasks but incur diminishing returns relative to their computational cost. These results highlight the importance of balancing accuracy with efficiency and suggest that mid-sized models constitute the practical "sweet spot" for zero-shot sentiment analysis on commodity hardware.

* *Corresponding author.*
E-mail addresses: aydin.gasimov@protonmail.com (Aydin Gasimov).
https://mcs.beu.edu.az/xxx.pdf

1. Introduction

The rapid progress of large language models (LLMs) has transformed natural language processing (NLP), enabling significant advances in tasks such as translation, summarization, and sentiment analysis [1; 2]. However, most state-of-the-art LLMs are extremely resource-intensive, with hundreds of billions of parameters requiring specialized hardware and substantial financial investment to operate at scale [3]. This reliance on massive centralized models raises critical questions about accessibility, cost, privacy, and sustainability [4].

Smaller, locally run LLMs—typically ranging from a few hundred million to a few billion parameters—present a compelling alternative. They can be executed on commodity hardware such as consumer GPUs, laptops, or edge devices [5], reducing dependency on cloud infrastructures. This not only lowers the barrier to entry for research groups, startups, and individual practitioners, but also dramatically reduces inference costs and energy consumption. Furthermore, local deployment addresses growing concerns around data privacy and security, especially in sensitive domains such as healthcare, finance, and legal services.

This shift stands in contrast to earlier generations of sentiment analysis methods. Traditional machine learning pipelines often relied on handcrafted features or static embeddings [6], while transformer-based models like BERT marked a turning point by introducing contextualized representations [7]. Yet, even BERT-style models require significant time and resources for fine-tuning before they are suitable for a private use-case. In comparison, recent developments in model distillation and pruning have expanded the feasibility of deploying compact LLMs in constrained environments without sacrificing competitive accuracy [8].

Despite these advantages, systematic evaluation of smaller LLMs on sentiment analysis remains limited. Most benchmarks emphasize performance on large, cloud-scale models [9], leaving open the question of how compact architectures perform across diverse datasets and domains. A rigorous comparison is necessary to understand their trade-offs in efficiency, accuracy, and robustness, and to determine whether such models can meet practical requirements without reliance on centralized infrastructures.

This paper aims to fill that gap by assessing the performance of locally run LLMs, up to 8B parameters in size, on a wide range of sentiment analysis datasets. By analyzing their strengths and limitations relative to both traditional approaches and large-scale models, we seek to provide a comprehensive perspective on their viability in research and applied contexts. Ultimately, this work contributes to the ongoing discourse on balancing efficiency, accessibility, and accuracy in the next generation of NLP systems.

2. Related Work

2.1 Pre-LLM Methods

Before the advent of large language models, sentiment analysis relied heavily on traditional machine learning and feature-engineering approaches. Early systems used bag-of-words or n-gram features combined with classifiers such as Support Vector Machines (SVMs) or logistic regression [6]. While effective for specific domains, these methods struggled to capture context, irony, and long-range dependencies in text.

The introduction of pretrained embeddings such asWord2Vec [10] and GloVe [11] marked a significant shift, enabling models to leverage distributional semantics. However, these static embeddings could not account for word meaning variation in different contexts (e.g., “bank” as a financial institution vs. riverbank).

A major breakthrough came with the transformer architecture [1] and contextualized embeddings, most notably BERT [7]. BERT and its variants (e.g., RoBERTa [12], DistilBERT [8], and ALBERT [13]) achieved state-of-the-art performance across many sentiment benchmarks by modeling bidirectional context. These models became widely adopted for text classification, including fine-grained and aspect-based sentiment analysis.

Despite their success, BERT-style models required substantial compute resources for fine-tuning and inference, limiting their practicality outside research or well-resourced industry labs. Moreover, their reliance on cloud deployment raised concerns about privacy and data confidentiality in domains such as healthcare and finance. These limitations paved the way for investigating smaller, more efficient models that could be deployed locally, while still retaining strong performance on core NLP tasks like sentiment analysis.

2.2 Sentiment Analysis with LLMs

Recent studies have begun to evaluate small language models (SLMs) as practical alternatives to resource-intensive LLMs. Pham et al. introduced SLM-Bench, the first large-scale benchmark dedicated to small language models, evaluating 15 models across 23 datasets and nine task categories [14]. Unlike earlier benchmarks that focused narrowly on accuracy, SLM-Bench integrates additional dimensions such as runtime, energy consumption, and CO2 emissions. This work demonstrates that some small models achieve competitive accuracy while offering markedly lower energy costs, highlighting their potential for sustainable deployment. However, the benchmark remains broad in scope: while it includes classification tasks such as sentiment analysis, it does not provide a detailed exploration of how SLMs handle the nuanced challenges of emotional tone, sarcasm, or aspect-based opinion mining.

Tan et al. proposed a learnware framework for organizing and deploying specialized SLMs across domains such as finance, healthcare, and mathematics [15]. This system outperformed individual SLMs and even surpassed large 70B models in domain-specific tasks, highlighting the potential of decentralized, privacy-preserving model reuse. However, its application to sentiment analysis remains largely unexplored.

Lepagnol et al. conducted a large-scale study on zero-shot classification, comparing models from 77M to 40B parameters across 15 datasets [16]. They showed that small models, when instruction-tuned and paired with suitable prompting strategies, can rival or surpass larger models. This finding reinforces the argument that model size is not the sole determinant of classification performance, particularly for sentiment-oriented datasets. Couto et al. examined compact multilingual LLMs on Iberian languages, introducing benchmarks that stress-test models in low-resource and end-user deployment scenarios [17]. Their results reveal persistent gaps for subtle linguistic phenomena like irony and sarcasm, suggesting that sentiment analysis in underrepresented languages remains an open challenge for compact models.

Koto et al. proposed a multilingual lexicon-based pretraining method for zero-shot sentiment analysis across 34 languages, including 25 low-resource ones [18]. Their approach outperformed GPT–3.5, BLOOMZ, and XGLM in many settings, underscoring that lexicon-augmented SLMs can generalize better across diverse linguistic contexts without relying on sentence-level annotations.

Liu et al. examined whether large language models actually possess sentiment sensitivity and are capable of consistently detecting emotional tone [19]. Their experiments revealed that while LLMs show basic sensitivity, their performance varies widely across datasets and model families. Misclassifications were especially common in cases involving subtle emotional cues such as irony or strongly contextual expressions. Importantly, the paper underscores that different architectures produce divergent outcomes even under identical testing conditions, suggesting that sentiment performance is far from a solved problem.

Together, these studies highlight efficiency, modularity, multilingual generalization, and sentiment awareness as key themes, yet none provide a systematic evaluation of small, locally run models on diverse sentiment datasets. Our work addresses this gap by benchmarking ≤8B models specifically for sentiment analysis.

3. Datasets Used

To ensure a comprehensive evaluation of compact large language models on sentiment analysis, we selected a diverse range of benchmark datasets capturing different challenges, from large-scale and fine-grained movie review corpora (IMDB, SST-2, SST-5) to domain-specific texts (FinancialPhraseBank), social media data with informal language and offensive content (Twitter Airline, OLID), emotion-oriented resources (Emotions), and aspect-based sentiment tasks (SentiHood). This diversity allows us to test not only polarity detection but also subtle affective cues and multi-entity reasoning, thereby reflecting both established benchmarks and realistic application scenarios. To establish a baseline of worst-case performance, we also calculated the probability of correct classification under random guessing. For an imbalanced dataset, this probability is not uniform across classes but instead depends on their relative frequencies. Specifically, if a dataset has K classes with class proportions $p_1, p_2, \ldots, p_k$, then the expected accuracy

of random guessing is given by [20]

$$P_{random} = \sum_{i=1}^{K} \quad p^2{}_i \ (1)$$

This formula captures the intuition that randomly guessing according to the class distribution yields higher expected accuracy when one or a few classes dominate. Finally, to contextualize model performance, we reported the strongest available State of the Art (SOTA) scores for each dataset (Table 1), which provide a practical upper bound on model capability.

- **IMDB Reviews Dataset** The IMDB Movie Reviews dataset is a widely used benchmark for sentiment analysis, containing 50,000 movie reviews labeled as either positive or negative. The dataset has an exact 50/50 split of positive and negative reviews and it was first introduced by Maas et al. in 2011. [21] Since this dataset is much larger than other datasets in our testing, we took a stratified sample of 10,000 reviews, preserving the 50/50 distribution. In our testing, we found that models perform slightly better on longer sentiments and slightly worse on shorter sentiments. We measured that the sample we took had near-identical mean and median values for review length to the original dataset.
- **Stanford Sentiment Treebank SST-5** The Stanford Sentiment Treebank SST-5 is a fine-grained sentiment analysis benchmark introduced by Socher et al. (2013) [22], constructed from the Rotten Tomatoes movie review corpus. The dataset was originally annotated across 25 distinct sentiment levels that were obtained using Amazon Mechanical Turk crowdworkers. For evaluation, the 25-way labels are typically collapsed into five categories (very negative, negative, neutral, positive, very positive), defining the widely used SST-5 task for sentence-level sentiment classification.
- **Stanford Sentiment Treebank SST-2** The Stanford Sentiment Treebank SST-2 is a binary sentiment classification benchmark introduced by Socher et al. (2013) [22] and popularized by the GLUE Benchmark [23]. The dataset was constructed by discarding the neutral class from the fine-grained annotations, leaving only positive and negative labels. It should be

noted that, while the SOTA scores reported in the table are based on the test set, our experiments were conducted on the validation set, since the gold test labels are withheld by the GLUE benchmark organizers.

- **FinancialPhraseBank** The FinancialPhraseBank is a domain-specific sentiment analysis dataset consisting of financial news sentences annotated as positive, negative, or neutral. It was introduced by Malo et al. in 2014 [24]. The sentences were labeled by a group of finance experts to ensure high-quality domain-relevant sentiment annotations.
- **Twitter Airline Sentiment Dataset** This dataset, introduced by Crowdflower in 2015 [25], contains tweets directed at major U.S. airlines, labeled as positive, neutral, or negative. Due to its social media origin, it introduces challenges such as informal language, sarcasm, and domain-specific vocabulary. It has become a benchmark for evaluating sentiment models in noisy, short-form text typical of Twitter data.
- **Emotions Dataset** The Emotions dataset is a fine-grained sentiment analysis benchmark containing English sentences annotated with six emotion categories (sadness, joy, love, anger, fear, surprise). It was introduced by Saravia et al. in 2018 [26], where it was proposed as a resource for evaluating emotion classification models beyond simple polarity detection
- **OLID (Offensive Language Identification Dataset)** The Offensive Language Identification Dataset (OLID) is a benchmark for offensive language detection in social media, annotated with a hierarchical three-level labeling scheme (offensive vs. not offensive; targeted vs. untargeted; and target category). It was introduced by Zampieri et al. (2019) [27] for the SemEval-2019 Task 6 (OffensEval) shared task, making it a widely used resource for studying abusive language classification. OLID is made up of 3 levels and we are using the first level which is the largest and most commonly used subset.
- **SentiHood** The SentiHood dataset is an aspect-based sentiment analysis (ABSA) benchmark focused on urban neighborhood discussions extracted from Yahoo! Answers. Each instance consists of a sentence mentioning one or more neighborhoods, annotated with aspect categories (e.g., price, safety, live, quiet) and their associated sentiment polarity (positive, negative, none). It was introduced by Saeidi et al. in 2016 [28].

Table 1. Random guessing probabilities and reported SOTA scores for benchmark sentiment analysis datasets.

Dataset	Random Guessing (%)	SOTA Score (%)
IMDB	50.00	96.21 [29]
SST-5	21.48	60.48 [30]
SST-2	50.02	97.9 [31]
FinancialPhraseBank	44.76	90.9 [32]
Twitter Airline	46.39	92.36 [33]
Emotions	24.40	88.5 [34]
OLID (Level A)	55.62	85.12 [35]
SentiHood	60.08	93.8 [36]

4. Large Language Models used

Large Language Models (LLMs) were organized into four categories according to their parameter size: large-scale models in the 7B-8B range, mid-sized models in the 3B–4B range,

smaller models slightly above or around 1B parameters and sub-1B models, the largest of which contains 0.5B parameters. Model size generally determines the overall performance potential of LLMs across diverse tasks, but larger models require greater memory capacity and typically operate more slowly, even on capable hardware. An additional aim of this study was to evaluate the practicality of newer, compact LLMs for sentiment analysis. To this end, several sub-1B models were selected and assessed for their performance characteristics. All tested models were quantized to Q8 (8-bit), except for Gemma3 1B QAT, which applies a specialized quantization-aware training method to achieve improved performance at Q4_0 (4-bit) quantization [37]. All models considered were non-reasoning variants. Although reasoning-augmented LLMs have demonstrated substantial gains on complex tasks, they also introduce significant latency and computational overhead. While such capabilities may be beneficial in sentiment analysis for scenarios without strict time constraints, they fall outside the scope of this paper.

One notable exception concerns the Qwen3 family, which achieves state-of-the-art results within its respective categories. The 1.7B, 4B, and 8B Qwen3 models are hybrid reasoning architectures, allowing reasoning to be toggled at inference. However, when tested in non-reasoning mode, their performance was unsatisfactory, and they were excluded from comparison to avoid misrepresenting typical use. Qwen3 0.6B was tested across all eight datasets but consistently failed to yield meaningful results, leading to its removal from tables and evaluations [38].

All models employed were instruction-tuned and evaluated in a zero-shot setting for sentiment analysis. No additional fine-tuning was applied beyond their publicly released versions.

4.1 Models in 7B to 8B Range

- **Llama 3 8B** Part of Meta's Llama 3 family, this 8B parameter model provides strong general-purpose performance across reasoning, comprehension, and generation tasks. It is widely adopted as a standard mid-sized baseline in research and practice. Llama 3 8B is one of the most widely used LLMs in the 8B range. While it does have a newer version that scores slightly higher on intelligence benchmarks, we found that Llama 3 8B is more reliable when it comes to sentiment analysis tasks than Llama 3.1 8B. [39]
- **Granite 3.2 8B** Developed by IBM, Granite 3.2 8B is an open foundation model optimized for enterprise and domain-specific workloads. It emphasizes trustworthy outputs and efficiency, making it well-suited for applied research settings. [40]
- **MiniCPM 4 8B** MiniCPM-4 is a lightweight, high-performance model developed by OpenBMB. The 8B variant balances efficiency with strong multilingual and reasoning capabilities. [41]
- **OLMoE-1B-7B-0125** Released by the Allen Institute for AI, this model follows a Mixture-of-Experts (MoE) design with 64 experts and 1B active parameters per forward pass from a 7B parameter pool [42]. Sparse expert routing reduces compute compared to dense 7B–8B models [43]. While MoE architectures are more common in >20B parameter models, AllenAI's OLMoE is one of the few implementations scaled down to the 7B range. The version we tested is an updated variant released in January of 2025.

4.2 Models in 3B to 4B Range

- **Gemma 3 4B** Released by Google DeepMind, Gemma 3 4B is designed for efficient deployment with high-quality text generation and alignment. It achieves strong performance relative to its size. This model is multi-modal and can work with visual inputs. [44]

- **Phi-4 Mini** A member of Microsoft's Phi series, Phi-4 Mini emphasizes compactness and reasoning efficiency. It is optimized for instructional tasks and educational applications. The model is 3 billion parameters in size. [45]
- **Llama 3.2 3B** A smaller variant in the Llama 3.2 family, the 3B model targets lower memory footprints while retaining competitive performance on common NLP. [39]

4.3 Models around 1B parameters

- **Liquid LFM-2 1.2B** A compact model optimized for research on quantization and low-resource inference. It is frequently employed in efficiency studies and resource-constrained deployment scenarios. [46]
- **Llama 3.2 1B** The smallest official member of the Llama 3.2 family, this 1B parameter variant is designed for experimentation on edge devices and constrained environments. [39]
- **Gemma 3 1B QAT** A quantization-aware trained (QAT) variant of Gemma 3, designed to preserve accuracy under aggressive compression, making it highly efficient for deployment on resource-limited systems. [44] [37]

4.4 Sub-1B parameter models

- **MiniCPM 4 0.5B** A 0.5B variant of MiniCPM, this model is tuned for speed and compact deployment while maintaining acceptable performance on standard NLP tasks. [41]
- **SmolLM 2 360M** Part of the SmolLM family, this 360M parameter model is designed as an extremely lightweight transformer for quick experimentation and edge-level deployment. [47]
- **Ernie 4.5 0.3B** Released by Baidu, Ernie 4.5 0.3B is a compact multilingual model integrating knowledge-enhanced pretraining with efficient inference performance. It is the smallest model in the family and the only dense (non-MoE) model. [48]
- **SmolLM 2 135M** The smallest model in the SmolLM 2 family, with 135M parameters, designed primarily for testing scalability and micro-deployment scenarios. This is the smallest model that we tested and it pushes the limits of usability at very little compute cost. [47]

5. Methodology

5.1 Running automation script

We automated evaluation with lightweight Python scripts that handle both dataset loading and model prompting. Each dataset was normalized to a minimal (text, label) format, with adaptations for tasks such as aspect-based sentiment analysis or multi-class emotion classification. The scripts support multiple file formats (CSV, JSONL, Parquet), ensuring consistency across heterogeneous datasets.

5.1.1 Prompts and Decoding

Each prompt was deliberately minimal, restricting the model to a closed label set. For example, binary sentiment tasks required exactly positive or negative, fine-grained SST-5 required a digit in {0, 1, 2, 3, 4}, and ABSA tasks required one of {positive, negative, none}. To enforce validity, outputs were post-processed with simple regex and keyword matching rules that coerce stray variants into the canonical label set.

5.1.2 Deterministic Decoding

All runs were performed with temperature fixed at T = 0, corresponding to greedy decoding. Given logits z_t at step t, the probability distribution is defined as

$$p_t(i) = \frac{exp(z_{t,i}/T)}{\sum_j exp(z_{t,j}/T)} \quad (2)$$

As T → 0, the distribution collapses to a point mass on the maximum logit, yielding

$$y_t = argmax_i z_{t,i} \quad (3)$$

This setting is equivalent to top-k sampling with k = 1 or nucleus sampling with p → 0. It guarantees reproducibility by eliminating randomness and ensures that model comparisons are strictly deterministic.

5.2 Evaluation Metrics

The models were evaluated using standard classification metrics [49]:

- **Accuracy:** $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$ (3)
- **Macro-F1 score:** $MacroF1 = \frac{1}{C}\sum_{i=1}^{C} \frac{2 \cdot Precision_i \cdot Recall_i}{Precision_i + Recall_i}$ (4)
- **Matthews Correlation Coefficient (MCC):** MCC is a correlation coefficient between the observed and predicted binary classifications. It takes into account true and false positives and negatives, producing a value between −1 and +1. A value of +1 indicates perfect prediction, 0 corresponds to random prediction, and −1 indicates total disagreement between prediction and ground truth. [50] Unlike accuracy or F1-score, MCC is considered a balanced measure because it remains informative even when the dataset is imbalanced. This makes it particularly suitable for sentiment classification tasks where some classes may be underrepresented.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (5)$$

- **Cohen's Kappa (κ):** Cohen's Kappa measures the agreement between predicted and true labels, adjusting for the possibility of agreement occurring by chance. It ranges from −1 (complete disagreement) to +1 (perfect agreement), with 0 representing chance-level performance. [51] The advantage of κ is that it penalizes models that achieve seemingly high accuracy simply by exploiting class imbalances. In multi-class sentiment analysis, this is especially valuable because it reflects how reliably the model performs across all categories, rather than being dominated by the most frequent class.

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \quad (6)$$

For the **SST-5 dataset**, the sentiment labels form an ordinal scale ranging from very negative to very positive. While the task is evaluated as classification, the ordered nature of the labels allows prediction errors to be meaningfully interpreted in terms of distance between categories. To capture the magnitude of these deviations, we additionally report standard regression-style error metrics that quantify absolute and squared differences between predicted and true labels [52]

- **Mean Absolute Error:** $MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$ (7)
- **Mean Squared Error:** $MAE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ (8)
- **Root Mean Squared Error:** $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$ (9)

5.3 Speed and Latency testing

It is widely recognized that larger language models (LLMs), which require greater memory capacity, also demand more computational resources during inference. However, architectural differences can further influence performance, allowing certain models to achieve advantages in speed. To account for these variations, we conducted measurements of speed and latency for several LLMs executed locally.

For sentiment analysis tasks, prompt processing speed is a more meaningful performance indicator than raw token throughput (tokens per second). This is because models must process long input sequences while often producing only one or two output tokens. Such processing is primarily constrained by the memory bandwidth of the RAM/VRAM allocated to the processor. To evaluate this dimension of performance, we constructed three dedicated datasets designed specifically for controlled prompt ingestion, rather than for accuracy evaluation.

Our experiments showed that response latency scales significantly with input length: for instance, when classifying long IMDB reviews, models often required more than ten times the processing time compared to short reviews. Accordingly, we defined three test sets with average sentence lengths of approximately 30, 150, and 1200 tokens. Sentences were primarily sampled from the IMDB dataset, as it provides the widest variation in input length, including reviews exceeding 1000 tokens.

All tests were performed using llama.cpp (Windows Vulkan backend, version 1.50.2). The hardware consisted of an AMD Radeon 890M integrated GPU with 8 GB of dedicated GPU memory (BIOS allocation) and 4 GB of shared GPU memory (32 GB total) with a rated bandwidth of 120 GB/s. The GPU driver version was 25.10.25.10-250825a-418637C, and the Vulkan API version was 1.4.315. Given that the tests were run on a laptop platform, fan profiles influenced power delivery. Stable performance was observed at the lowest fan setting, where power draw remained at approximately 10 W with minimal spikes. All benchmarks were therefore conducted under these conditions to avoid variability from power fluctuations.

It is important to note that these results are specific to GPUs operating under Vulkan. Models optimized for other backends such as MLX, CUDA, or OpenVINO (e.g., MiniCPM-4) may exhibit different performance characteristics. Furthermore, while CPU inference on the same test system (Ryzen AI 9 HX 370) achieved comparable token throughput, prompt processing was three to four times slower, rendering CPU execution impractical for high-volume sentiment analysis workloads.

6. Accuracy Results

The evaluation results are presented across ten tables, with additional breakdowns for SST-5 and the Emotions dataset to account for their fine-grained label structures. Each table reports five complementary metrics-accuracy, macro-F1, micro-F1, Matthews Correlation Coefficient (MCC), and Cohen's κ, chosen to capture both overall correctness and robustness under class imbalance.

The highest score within each metric group is highlighted in bold. In some cases, particularly with the smallest models, the "best" result within a group may reflect the least inadequate performance rather than genuine effectiveness, but these values are retained for completeness and comparability.

6.1 IMDB Dataset

On binary sentiment classification of long-form reviews, larger models consistently outperformed compact ones. OLMoE-1B-7B achieved the strongest performance (accuracy 95.3%), closely followed by MiniCPM-4 8B and IBM Granite 3.2 8B. Among smaller models, Liquid LFM-2 1.2B and Gemma 3 1B QAT demonstrated competitive accuracy (≥87%), indicating that carefully optimized 1B–1.2B models can approach near-state-of-the-art results on balanced datasets. Among the smallest models, Ernie 4.5 0.3B greatly outperformed its peers and came close to much larger models, suggesting that this dataset could've been part of its training data. (Table 2)

Table 2. IMDB Dataset results

Model	Accuracy	Macro-F1	MCC	Cohen's κ
SmolLM2 135M	0.617	0.585	0.281	0.234
Ernie 4.5 0.3B	**0.872**	**0.872**	**0.749**	**0.744**
SmolLM2 360M	0.796	0.791	0.620	0.592
MiniCPM4 0.5B	0.635	0.583	0.381	0.270
Gemma 3 1B QAT	0.874	0.874	0.751	0.748
Llama 3.2 1B Instruct	0.826	0.824	0.664	0.652
Liquid LFM2 1.2B	**0.921**	**0.921**	**0.842**	**0.842**
Llama 3.2 3B Instruct	0.694	0.688	0.403	0.388
Phi-4 Mini	**0.925**	**0.925**	**0.850**	**0.850**
Gemma 3 4B	0.896	0.895	0.803	0.792
OLMoE-1B-7B-0125	**0.953**	**0.953**	**0.907**	**0.906**
Llama 3 8B Instruct	0.917	0.917	0.837	0.834
IBM Granite 3.2 8B	0.934	0.934	0.869	0.868
MiniCPM4 8B	0.945	0.945	0.891	0.890

6.2 Stanford Sentiment Treebank SST-5

SST-5. Fine-grained sentiment classification proved especially challenging. Even larger models struggled, with the best result (Gemma 3 4B, accuracy 52.1%) falling short of the reported SOTA (Table 3). Smaller models frequently collapsed predictions into a subset of classes, yielding poor macro-F1 and inflated per-class sparsity (Table 4). This highlights the difficulty of fine-grained polarity detection in zero-shot settings, where subtle contextual cues are often missed.

Table 3. SST-5 Overall results

Model	Accuracy	MAE	MSE	RMSE	Cohen's κ
SmolLM2 135M	0.165	**1.28**	**2.37**	**1.54**	-0.00869
Ernie 4.5 0.3B	0.127	2.05	5.96	2.44	0.000438
SmolLM2 360M	**0.176**	1.95	5.5	2.35	-0.0118
MiniCPM4 0.5B	0.167	1.80	4.88	2.21	**0.0132**
Gemma 3 1B QAT	0.255	1.23	2.51	1.58	**0.0626**
Llama 3.2 1B Instruct	0.187	1.78	4.82	2.20	0.0110
Liquid LFM2 1.2B	**0.296**	**1.12**	**2.16**	**1.47**	0.0568
Llama 3.2 3B Instruct	0.330	0.865	1.33	1.15	0.164
Phi-4 Mini	0.455	0.623	0.799	0.894	0.295
Gemma 3 4B	**0.521**	**0.543**	**0.689**	**0.830**	**0.366**
OLMoE-1B-7B-0125	0.316	1.065	2.015	1.419	0.108
Llama 3 8B Instruct	0.426	0.640	**0.781**	**0.884**	0.244
IBM Granite 3.2 8B	**0.496**	**0.588**	**0.781**	**0.884**	**0.377**
MiniCPM4 8B	0.437	0.656	0.854	0.924	0.306

Table 4. SST-5 per-class F1 scores

Model	V. Negative	Negative	Neutral	Positive	V. Positive
SmolLM2 135M	0.126	0.0453	**0.283**	0	0
Ernie 4.5 0.3B	0.224	0	0.00513	0	0
SmolLM2 360M	0.0488	0.0673	0	0	**0.294**
MiniCPM4 0.5B	**0.246**	**0.101**	0	**0.175**	0.0816
Gemma 3 1B QAT	**0.286**	0.0529	0	**0.405**	0
Llama 3.2 1B Instruct	0.235	0.296	0	0.0372	0
Liquid LFM2 1.2B	0.238	**0.456**	**0.152**	0.175	**0.0242**
Llama 3.2 3B Instruct	0.397	0.109	**0.357**	0.462	0.0933
Phi-4 Mini	**0.44**	0.543	0.323	0.571	0
Gemma 3 4B	0.345	**0.647**	0.265	**0.596**	**0.326**
OLMoE-1B-7B-0125	0.007	0.327	0.099	0.406	0.311
Llama 3 8B Instruct	0.0685	**0.64**	0.345	0.415	0.0675
IBM Granite 3.2 8B	0.514	0.319	0.375	**0.592**	**0.656**
MiniCPM4 8B	**0.536**	0.0455	**0.412**	0.581	0.516

6.3 Stanford Sentiment Treebank SST-2

In contrast, binary sentiment classification on SST-2 yielded strong results across model families. Phi-4 Mini (93.2%), Gemma 3 4B (93.0%), and IBM Granite 3.2 8B (93.3%) performed comparably, with OLMoE-1B-7B narrowly leading at 94.0%. Even lightweight 1B models such as Liquid LFM-2 (90.4%) showed reliable performance (Table 5), reflecting the relative simplicity of binary sentiment tasks when compared to SST-5.

Table 5. SST-2 results

Model	Accuracy	Macro-F1	MCC	Kohen's κ
SmolLM2 135M	0.766	0.764	0.547	0.534
Ernie 4.5 0.3B	**0.818**	**0.814**	**0.671**	**0.637**
SmolLM2 360M	0.620	0.566	0.370	0.251
MiniCPM4 0.5B	0.795	0.790	0.630	0.592
Gemma 3 1B QAT	0.882	0.882	0.764	0.764
Llama 3.2 1B Instruct	0.878	0.878	0.772	0.758
Liquid LFM2 1.2B	**0.904**	**0.904**	**0.809**	**0.808**
Llama 3.2 3B Instruct	0.686	0.659	0.470	0.378
Phi-4 Mini	**0.932**	**0.932**	**0.867**	**0.865**

Gemma 3 4B	0.930	0.930	0.866	0.860
OLMoE-1B-7B-0125	**0.940**	**0.940**	**0.883**	**0.881**
Llama 3 8B Instruct	0.497	0.342	0.0746	0.0111
IBM Granite 3.2 8B	0.933	0.933	0.874	0.867
MiniCPM4 8B	0.921	0.921	0.847	0.842

6.4 FinancialPhraseBank

Domain-specific sentiment analysis in finance revealed a clear stratification. MiniCPM-4 8B achieved the highest accuracy (79.1%), followed closely by Phi-4 Mini and Gemma 3 4B. While sub-1B models struggled, Liquid LFM-2 (63.0%) and Llama 3.2 1B (63.0%) provided mid-tier results (Table 6), suggesting limited domain transfer without fine-tuning.

Table 6. FinancialPhraseBank results

Model	Accuracy	Macro-F1	MCC	Cohen's κ
SmolLM2 135M	0.295	0.213	0.119	0.0226
Ernie 4.5 0.3B	0.309	0.291	**0.217**	0.141
SmolLM2 360M	**0.589**	**0.447**	0.205	**0.201**
MiniCPM4 0.5B	0.328	0.323	0.19	0.115
Gemma 3 1B QAT	0.471	0.491	0.352	0.259
Llama 3.2 1B Instruct	**0.63**	0.421	0.224	0.159
Liquid LFM2 1.2B	**0.63**	**0.595**	**0.397**	**0.383**
Llama 3.2 3B Instruct	0.641	0.448	0.263	0.233
Phi-4 Mini	0.735	**0.738**	**0.586**	**0.566**
Gemma 3 4B	**0.744**	0.712	0.555	0.546
OLMoE-1B-7B-0125	0.738	0.727	0.526	0.526
Llama 3 8B Instruct	0.659	0.492	0.312	0.269
IBM Granite 3.2 8B	0.647	0.448	0.277	0.236
MiniCPM4 8B	**0.791**	**0.758**	**0.608**	**0.597**

6.5 Twitter Airline Sentiment

Social media sentiment classification exposed substantial model variance. While compact models such as Ernie 4.5 (72.6%) achieved respectable results, stronger performance was seen in Phi-4 Mini (80.9%) and Gemma 3 4B (81.1%). Performance degradations in some larger models (e.g., Llama 3 8B, 37.7%) underscore the instability of zero-shot inference on noisy, sarcasm-rich text, as well as, the models' ability to follow instructions without defaulting to standard responses. (Table 7)

Table 7. Twitter Airline Sentiment results

Model	Accuracy	Macro-F1	MCC	Cohen's κ
SmolLM2 135M	0.202	0.138	0.0766	0.0213
Ernie 4.5 0.3B	**0.726**	**0.515**	**0.464**	**0.413**
SmolLM2 360M	0.237	0.236	0.0771	0.0437
MiniCPM4 0.5B	0.672	0.485	0.307	0.282
Gemma 3 1B QAT	0.715	0.496	0.496	0.459
Llama 3.2 1B Instruct	0.706	**0.598**	**0.510**	**0.491**
Liquid LFM2 1.2B	**0.742**	0.542	0.500	0.447
Llama 3.2 3B Instruct	0.212	0.118	0.0103	0.000372
Phi-4 Mini	0.809	**0.751**	**0.656**	**0.651**
Gemma 3 4B	**0.811**	0.726	0.640	0.632
OLMoE-1B-7B-0125	0.753	0.576	0.0199	0.00526
Llama 3 8B Instruct	0.377	0.415	0.222	0.145
IBM Granite 3.2 8B	**0.791**	**0.747**	**0.639**	**0.631**
MiniCPM4 8B	0.329	0.374	0.270	0.128

6.6 Emotions Dataset

Multi-class emotion detection presented another challenging benchmark. IBM Granite 3.2 8B emerged as the strongest performer (accuracy 56.6%), significantly outperforming most smaller models. Among mid-sized candidates, Phi-4 Mini and Gemma 3 4B performed competitively (Table 8). Sub-1B models largely failed to capture the diversity of emotional categories, frequently defaulting to a small subset of labels (Table 9) .

Table 8. Emotions Dataset Overall results

Model	Accuracy	Macro-F1	MCC	Cohen's κ
SmolLM2 135M	0.238	0.109	0.0182	0.0149
Ernie 4.5 0.3B	0.137	0.136	0.0189	0.0156
SmolLM2 360M	**0.289**	0.0748	-0.0145	-0.00138
MiniCPM4 0.5B	0.168	0.116	0.0181	0.0133
Gemma 3 1B QAT	**0.198**	0.123	**0.107**	**0.0695**
Llama 3.2 1B Instruct	0.183	**0.153**	0.0734	0.0491
Liquid LFM2 1.2B	0.113	0.107	0.0112	0.00884
Llama 3.2 3B Instruct	0.354	0.354	0.277	0.246
Phi-4 Mini	0.464	0.442	0.349	0.334
Gemma 3 4B	**0.465**	**0.447**	**0.375**	**0.351**
OLMoE-1B-7B-0125	0.165	0.114	0.043	0.03
Llama 3 8B Instruct	0.395	0.380	0.298	0.275
IBM Granite 3.2 8B	**0.566**	**0.477**	**0.434**	**0.428**
MiniCPM4 8B	0.403	0.354	0.269	0.252

Table 9. Emotions Dataset per-class F1 scores

Model	Sadness	Joy	Love	Anger	Fear	Surprise
SmolLM2 135M	0.41	0.188	0	0	0	0.0588
Ernie 4.5 0.3B	0.211	0.135	0.108	**0.102**	**0.192**	0.0696
SmolLM2 360M	**0.449**	0	0	0	0	0
MiniCPM4 0.5B	0.251	**0.216**	**0.146**	0	0.00844	0.0769
Gemma 3 1B QAT	0	**0.204**	**0.19**	**0.301**	0.0129	0.0274
Llama 3.2 1B Instruct	0.0734	0.0992	0.178	0.271	0.161	**0.138**
Liquid LFM2 1.2B	**0.157**	0.0497	0.0209	0.137	**0.209**	0.0661
Llama 3.2 3B Instruct	0.424	0.236	0.28	0.523	0.348	0.311
Phi-4 Mini	0.521	0.443	**0.344**	0.467	**0.548**	0.33
Gemma 3 4B	**0.538**	**0.467**	0.323	**0.544**	0.414	**0.396**
OLMoE-1B-7B-0125	0.0656	0.0586	0.0495	0.253	0.219	0.0351
Llama 3 8B Instruct	0.23	0.496	0.336	0.433	**0.446**	0.343
IBM Granite 3.2 8B	**0.617**	**0.674**	**0.343**	**0.525**	0.336	**0.365**
MiniCPM4 8B	0.459	0.379	0.325	0.415	0.4	0.146

6.7 OLID

Offensive language detection (Level A) was highly demanding for compact models. All sub-1.2B models failed to produce meaningful outputs, often defaulting to majority-class predictions that inflated accuracy but yielded near-zero κ and MCC. For example, SmolLM2 135M and 360M appeared to reach 72% accuracy, but with $\kappa \approx 0$, showing no real agreement beyond chance. Reliable performance only emerged with larger architectures: IBM Granite 3.2 8B (76.8%) led the group, followed by MiniCPM-4 8B (73.9%) and Llama 3 8B (70.4%). Mid-sized models such as Gemma 3 4B (68.2%) and Phi-4 Mini (71.2%) showed competitive results with balanced agreement, but anything below 1B parameters was essentially unusable (Table 10).

Table 10. OLID results

Model	Accuracy	Macro-F1	MCC	Cohen's κ
SmolLM2 135M	0.719	0.430	**0.0209**	**0.00666**
Ernie 4.5 0.3B	0.448	**0.443**	0.00853	**0.00666**
SmolLM2 360M	**0.722**	0.419	0	0
MiniCPM4 0.5B	0.282	0.224	0.0134	0.00127
Gemma 3 1B QAT	**0.559**	**0.549**	**0.208**	**0.173**
Llama 3.2 1B Instruct	0.308	0.281	-0.024	-0.00857
Liquid LFM2 1.2B	0.278	0.218	0	0
Llama 3.2 3B Instruct	0.539	0.539	0.304	0.214
Phi-4 Mini	**0.712**	0.558	0.161	0.144
Gemma 3 4B	0.682	**0.671**	**0.445**	**0.385**
OLMoE-1B-7B-0125	0.598	0.588	0.285	0.24
Llama 3 8B Instruct	0.704	0.683	0.418	0.387
IBM Granite 3.2 8B	**0.768**	**0.715**	0.430	0.430
MiniCPM4 8B	0.739	0.713	**0.459**	**0.438**

6.8 SentiHood

Aspect-based sentiment analysis proved highly sensitive to model capacity. Gemma 3 4B attained the strongest result (84.4%), with Phi-4 Mini close behind (78.8%). Sub-1B models displayed frequent misclassifications, and even some larger models underperformed (e.g., IBM Granite 3.2 8B at 60.0%), suggesting that ABSA remains a non-trivial challenge without explicit aspect-level supervision (Table 11).

Table 11. Sentihood dataset results

Model	Accuracy	Macro-F1	MCC	Cohen's κ
SmolLM2 135M	0.389	0.257	0.129	0.082
Ernie 4.5 0.3B	0.609	0.399	0.203	0.257
SmolLM2 360M	0.621	0.410	0.260	0.309
MiniCPM4 0.5B	**0.782**	**0.492**	**0.319**	**0.476**
Gemma 3 1B QAT	0.072	0.060	0.056	0.017
Llama 3.2 1B Instruct	**0.463**	**0.370**	**0.251**	**0.207**
Liquid LFM2 1.2B	0.354	0.261	0.179	0.112
Llama 3.2 3B Instruct	0.000	0.000	0.000	0.000
Phi-4 Mini	0.788	0.550	0.446	0.571
Gemma 3 4B	**0.844**	**0.559**	**0.466**	**0.659**
OLMoE-1B-7B-0125	**0.637**	**0.476**	0.33	**0.364**
Llama 3 8B Instruct	0.086	0.074	0.056	0.016
IBM Granite 3.2 8B	0.600	0.435	**0.340**	0.318
MiniCPM4 8B	0.495	0.337	0.267	0.210

Summary: Across datasets, the accuracy results reveal that the strongest overall performance did not come from the largest models, but rather from the mid-sized 3–4B range. In particular, Gemma 3 4B and Phi-4 Mini consistently achieved the highest or near-highest scores, often surpassing the 7–8B models on tasks such as SST-2, Twitter Airline, and Sentihood, while remaining highly competitive even on more demanding datasets like Emotions and FinancialPhraseBank. Larger models such as MiniCPM-4 8B and Granite 3.2 8B retained an advantage on specific benchmarks—notably IMDB and certain fine-grained emotion classification tasks—but their lead was neither consistent nor decisive. By contrast, models below 1B parameters generally failed to generalize, with their performance frequently collapsing to

trivial label distributions despite occasional isolated successes. Taken together, these results indicate that the most reliable balance of accuracy across diverse sentiment analysis settings is achieved by compact mid-sized models, with 7–8B systems offering marginal improvements only in select cases and at substantially higher computational cost.

7. Analyzing Model Failures

During evaluation, several lightweight instruction-tuned models systematically failed to comply with the classification directive (e.g., Table 11, Llama 3.2 3B Instruct). Instead of generating the expected single-token output such as positive, negative, or none, these models frequently reproduced fragments of the instruction prompt itself (e.g., "Please respond with one of the following words..."). Extending the decoding budget from a single token to 8–25 tokens did not resolve the issue; rather, it allowed the model to continue paraphrasing or elaborating on the instructions. In these cases, the parser could not extract a valid label, causing predictions to default to none. Similar behaviors have been reported in prior work, where instruction-tuned chat models echo or re-interpret prompts when evaluated outside of their intended template format [53; 54].

This failure mode reflects a broader mismatch between completion-style evaluation and the way many instruction-tuned models have been optimized. Chat-oriented models are typically trained and deployed under specific templates, where reinforcement from human feedback rewards polite, verbose responses. When these templates are not used, models may revert to their safer defaults of restating or expanding the given instructions, rather than emitting a constrained label [55]. Prior studies on prompt sensitivity have shown that small variations in formatting, role definitions, or system instructions can lead to large shifts in accuracy, even with identical underlying models [56].

A second, distinct failure pattern emerged in fine-grained sentiment classification tasks such as SST-5 and the Emotions dataset. Here, smaller models often ignored parts of the label space and collapsed predictions into only a few categories. Despite repeated adjustments to the prompt format, the models consistently defaulted to high-frequency or semantically safe labels such as positive or neutral. This tendency aligns with documented issues of label bias and surface form competition, where language models exhibit strong priors toward particular verbalizers and over-rely on frequent tokens at the expense of rare ones [57; 58]. The result was inflated sparsity across classes, leading to poor macro-F1 scores even when raw accuracy appeared less affected.

These observations highlight that failures were not arbitrary but arose from identifiable mechanisms: (i) instruction-following models misinterpreting prompts when stripped from their native templates, and (ii) smaller models lacking the capacity to fully represent fine-grained distinctions, leading to class collapse. Importantly, the use of a consistent prompt across models and datasets in this study ensured comparability, even though prompt design itself is a recognized source of variance in LLM evaluation [57]. Future work may explore mitigations such as contextual calibration, constrained decoding, or template alignment, but the uniform prompting strategy adopted here provides a fair and controlled baseline for cross-model analysis.

8. Normalized Accuracy Metric

A challenge in evaluating the performance of large language models (LLMs) across heterogeneous sentiment analysis datasets lies in the fact that raw accuracy is not directly

comparable. This stems from two issues. First, the baseline accuracy achievable by random guessing varies widely depending on the number of classes and their distribution. For example, in the IMDB dataset, a binary classification task, random guessing achieves 50% accuracy, whereas in the five-class SST-5 dataset the baseline is only 20% (Figure 1). Second, the attainable performance upper bound also differs between datasets: while state-of-the-art (SOTA) models can exceed 95% accuracy on IMDB, the best reported results on SST-5 remain around 60%. Although it is theoretically possible for other models to surpass these scores, they are typically obtained through task-specific fine-tuning. By contrast, the smaller LLMs evaluated here are tested in a zero-shot setting, where such levels of performance are not realistically anticipated.

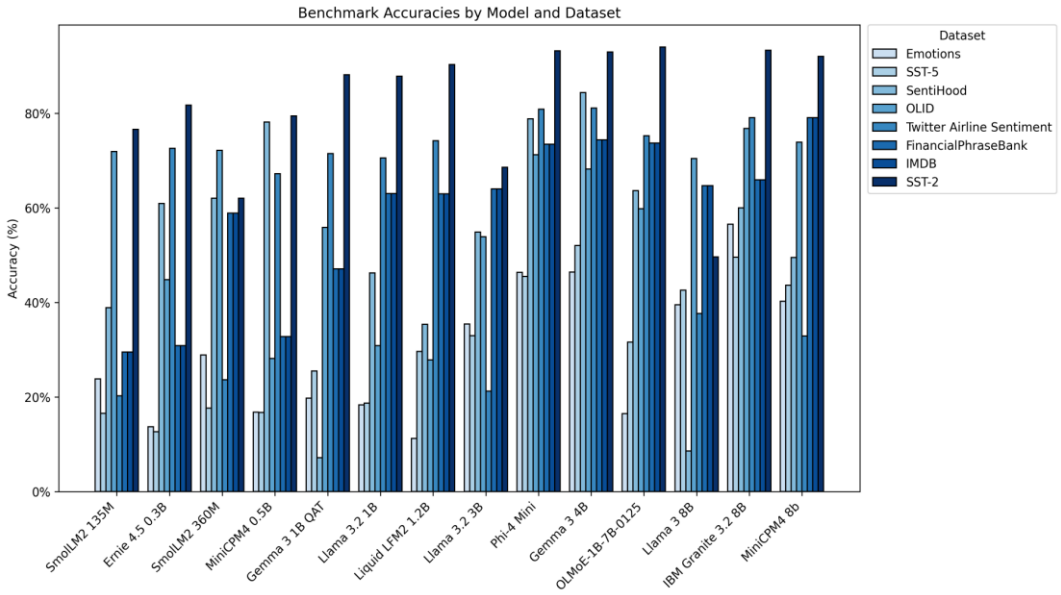

Fig. 1. Model accuracies across dataset before normalization

Existing metrics such as Matthews Correlation Coefficient (MCC) or Cohen's Kappa attempt to account for chance agreement, but they still suffer from comparability issues. For instance, obtaining an MCC of 0.4 on a complex five-class dataset may indicate stronger relative performance than achieving 0.8 on a balanced binary dataset. Thus, these measures do not provide a unified scale across tasks with different inherent difficulties. To address this, we introduce a **normalized accuracy metric**. For each dataset, we define a performance range between two reference points:

- **Lower bound:** the probability of correct classification by random guessing (A_{rand}), computed as the weighted sum of class proportions.
- **Upper bound:** the accuracy of the strongest available SOTA model (A_{SOTA}) reported in the literature.

The observed accuracy of a model on the dataset (A_{model}) is then linearly rescaled into a 0–100 scale:

$$NormAcc = \frac{A_{model} - A_{rand}}{A_{SOTA} - A_{rand}} \times 100\% \quad (10)$$

In this formulation, a model matching random guessing receives a score of 0, while a model achieving the SOTA accuracy receives 100. Scores below random guessing yield negative values, penalizing cases where the model fails to comply with the evaluation format under the given

prompt template (Figure 2). For example, in some instances models produced long descriptive outputs instead of the required single-word classification. This treatment ensures that poor performance on a single dataset meaningfully reduces the model's overall average (Figure 3).

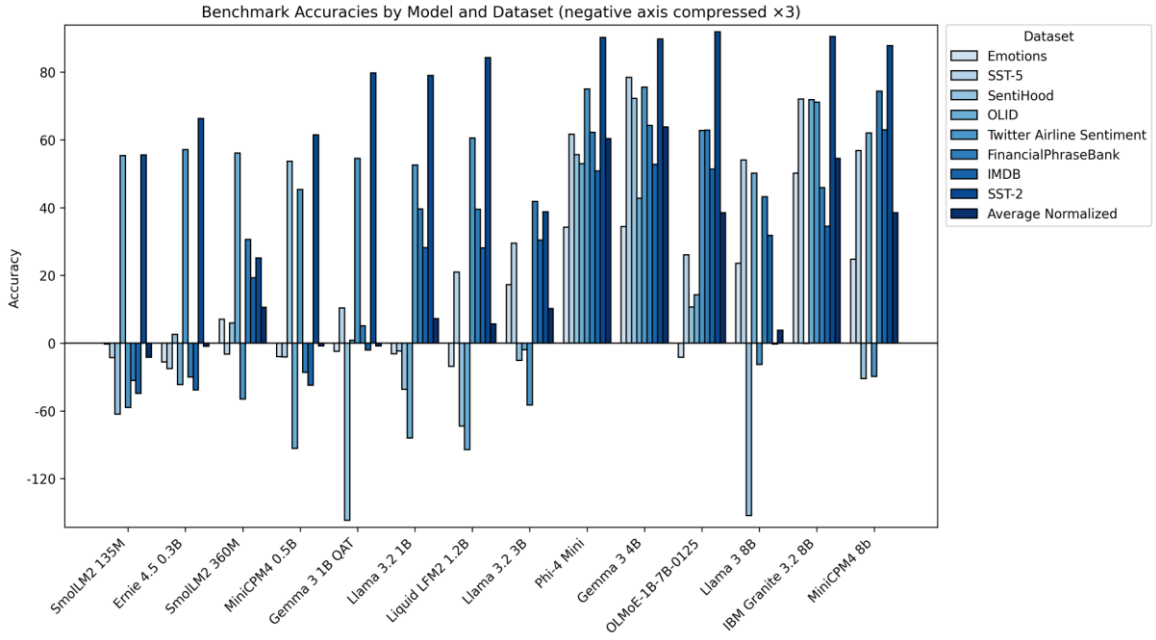

Fig. 2. Model accuracies across dataset after normalization

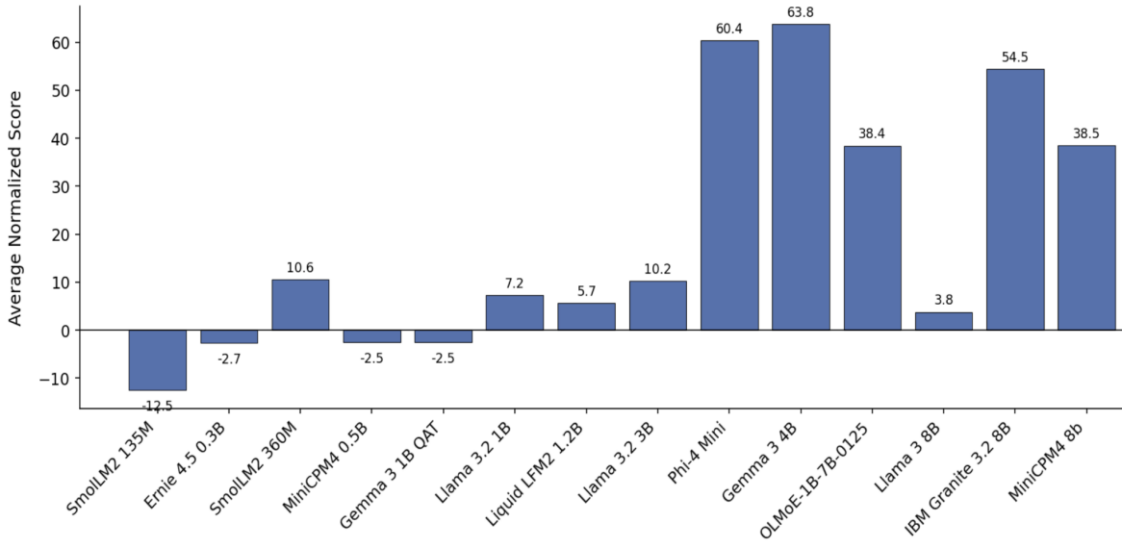

Fig. 3. Average normalized accuracy scores with penalization

At the same time, to avoid overly harsh penalization, we also consider an alternative version of the metric where scores below random guessing are clipped to zero (Figure 4). This second view treats severe underperformance as contributing no positive value, rather than further lowering the average, and can be more appropriate when the goal is to prevent outliers from disproportionately affecting results (Figure 5).

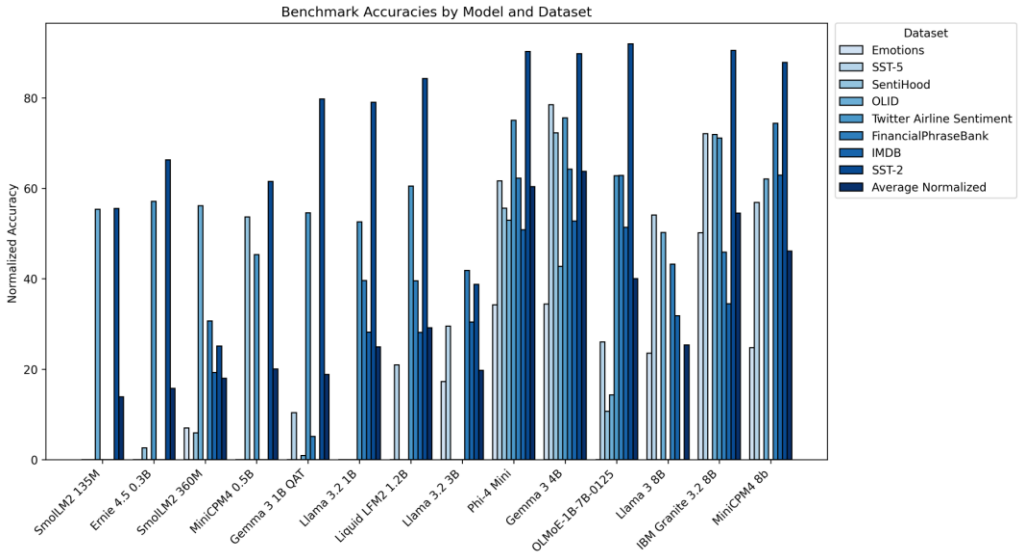

Fig. 4. Model accuracies across dataset after normalization with no penalty

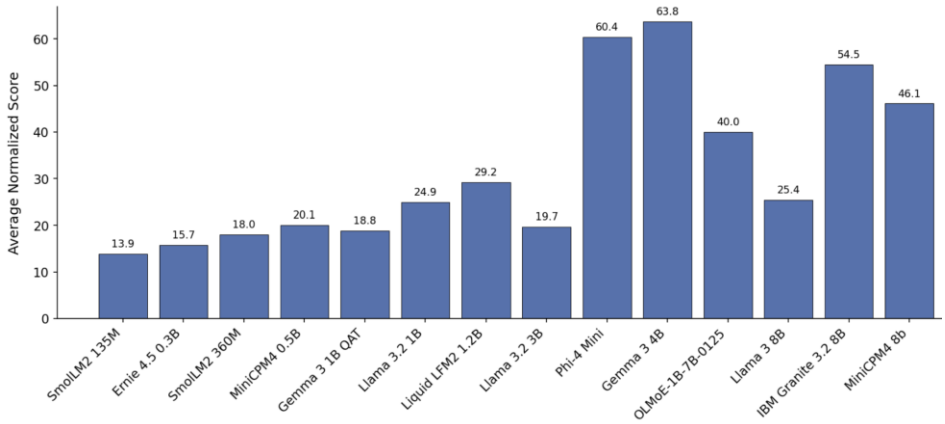

Fig. 5. Average normalized accuracy scores with no penalty

By normalizing in this way, we produce a comparable metric across datasets regardless of class imbalance or intrinsic task difficulty. For example, performance on IMDB (50% baseline, 95% SOTA) and SST-5 (20% baseline, 60% SOTA) are mapped to the same 0– 100 scale, despite the raw accuracies being on very different ranges. Crucially, this shared scale enables us to calculate the **average normalized score across all eight datasets** for each model (Figures 3 and 5), yielding a single interpretable performance figure that reflects its relative success across tasks of varying complexity. This averaging step is the central motivation for introducing the normalization, as it allows systematic comparison of models in a way that raw accuracy cannot.

9. Speed and Latency

We measured the average time per review (in seconds) and reviews per minute (RPM) after running the models through each of the 3 datasets and recorded the results (Tables 12 and 13). We also added the memory usage at a standard 4096 token context window for each of the models (Table 13). The best result in each category is highlighted in bold, though unsurprisingly the smallest model in each group is the fastest.

Table 12. Time per review (seconds) and Reviews per Minute for reviews at 30-token and 150-token ranges.

Model	Time(30t)	RPM(30t)	Time(150t)	RPM(150t)
Llama 3 8B Instruct	0.564	106.34	1.515	39.61
MiniCPM4 8B	0.674	89.00	1.829	32.80
IBM Granite 3.2 8B	0.584	102.67	1.866	32.16
OLMoE-1B-7B-0125	**0.233**	**257.38**	**0.382**	**157.08**
Gemma 3 4B	0.238	252.41	0.442	135.73
Phi-4 Mini Instruct	0.223	268.84	0.421	142.49
Llama 3.2 3B Instruct	**0.194**	**309.37**	**0.361**	**166.35**
Liquid LFM2 1.2B	0.158	380.74	0.231	260.08
Llama 3.2 1B Instruct	0.097	618.23	0.154	388.83
Gemma 3 1B QAT	**0.092**	**653.20**	**0.130**	**462.58**
MiniCPM4 0.5B	0.060	999.75	0.087	685.95
SmolLM2 360M Instruct	0.055	1091.38	0.076	788.47
ERNIE 4.5 0.3B	0.050	1192.06	0.073	827.57
SmolLM2 135M Instruct	**0.044**	**1370.73**	**0.055**	**1098.64**

Table 13. Time per review (seconds) and Reviews per Minute for reviews at 1200-token range and the memory footprint of the models (with 4096 token context window)

Model	Time(1200t)	RPM(1200t)	Memory footprint
Llama 3 8B Instruct	11.886	5.05	8.54 GB
MiniCPM4 8B	15.202	3.95	8.7 GB
IBM Granite 3.2 8B	15.383	3.90	8.68 GB
OLMoE-1B-7B-0125	**2.318**	**25.89**	**7.36 GB**
Gemma 3 4B	2.279	26.33	4.98 GB
Phi-4 Mini Instruct	2.293	26.17	4.08 GB
Llama 3.2 3B Instruct	**2.033**	**29.51**	**3.42 GB**
Liquid LFM2 1.2B	1.090	55.03	1.25 GB
Llama 3.2 1B Instruct	0.790	75.84	1.32 GB
Gemma 3 1B QAT	**0.569**	**105.42**	**720.43 MB**
MiniCPM4 0.5B	0.460	130.45	542.74 MB
SmolLM2 360M Instruct	0.449	133.55	386.4 MB
ERNIE 4.5 0.3B	0.359	167.21	385.79 MB
SmolLM2 135M Instruct	**0.244**	**246.12**	**144.81 MB**

We also construct graphs that visualize how gains in prompt processing speeds diminish beyond a certain threshold as model sizes are reduced (Figures 6 and 7). These visualizations make it clear that while smaller models initially provide substantial improvements in throughput, the rate of return gradually decreases, revealing a point at which further downsizing yields only marginal benefits.

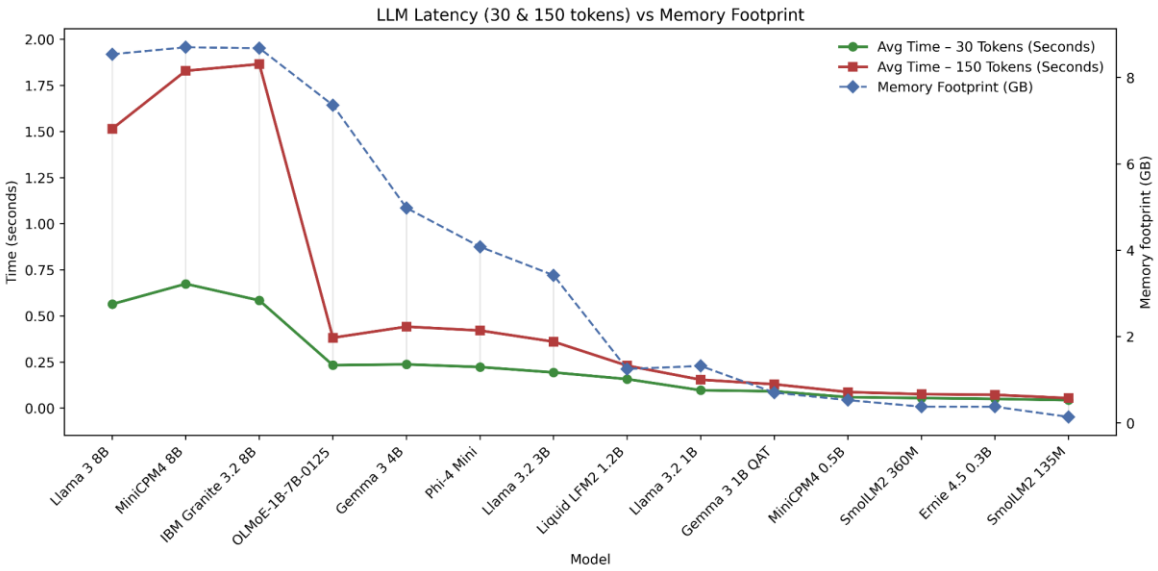

Fig. 6. Latency and Memory usage (30 tokens and 150 tokens)

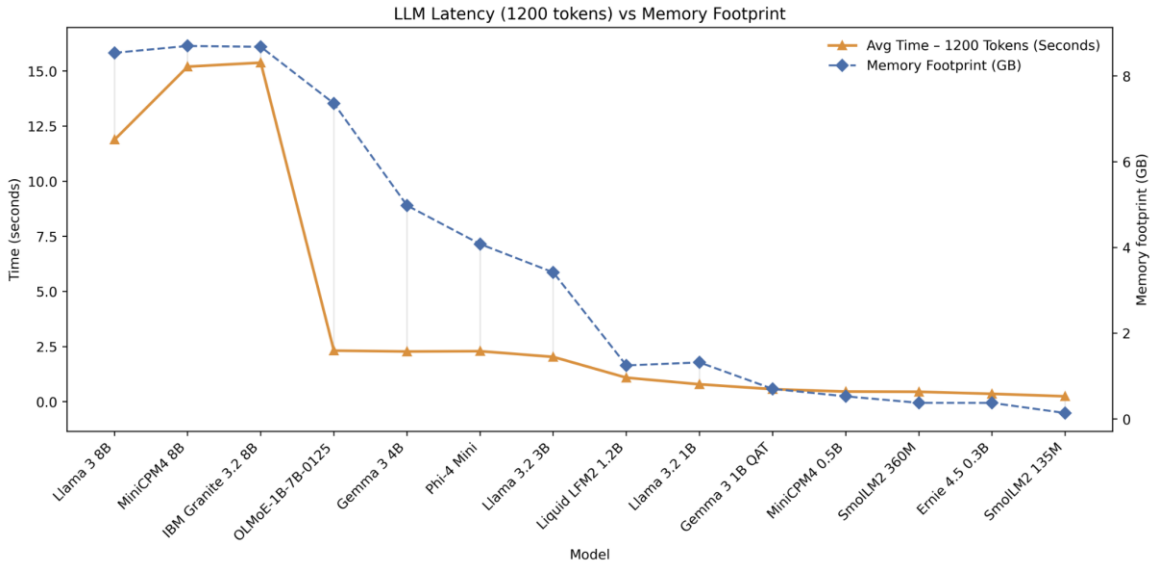

Fig. 7. Latency and Memory usage (1200 tokens)

The experimental results clearly demonstrate that inference latency is strongly determined by input length across all model sizes. For instance, Gemma-3 4B requires 0.238 s, 0.442 s, and 2.279 s to process inputs of approximately 30, 150, and 1200 tokens, respectively, confirming the near-linear growth of processing time with sequence length. At the same time, the relationship between latency and model size is less straightforward. While smaller models achieve faster processing speeds, the relative advantages diminish considerably once models fall below the 1B parameter range. As an illustration, SmolLM2-135M reaches 246 reviews per minute (RPM) at 1200 tokens, compared with 26.33 RPM for Gemma-3 4B, corresponding to a ≈ 9.3× improvement despite a ≈ 35× reduction in parameters. At shorter sequence lengths, this gap narrows further, with the speedup dropping to ≈ 5.4× at 30 tokens. These findings indicate that the practical gains from extremely small models are partially offset by their reduced accuracy, as shown in earlier evaluation sections.

The largest models tested (7–8B parameters) exhibit a pronounced increase in latency at long sequence lengths. At 1200 tokens, these models require approximately 12–15 s per review, compared with 2–2.3 s for 3–4B models and less than one second for 1B models. This disproportionate slowdown is explained by memory pressure: the observed memory footprints of ~8.5–8.7 GB exceed the 8 GB of dedicated VRAM available on the test hardware, leading to reliance on shared system memory and reduced throughput. Consequently, caution is warranted when selecting models that approach the memory capacity of the system, since longer prompts can easily push them over the threshold and result in severe performance degradation. Thus, performance in this regime arises not solely from parameter count, but also from hardware limitations.

An additional observation is that the Llama family exhibits slightly faster performance relative to other models in their respective parameter ranges. For example, Llama 3 8B Instruct processes 1200-token inputs more efficiently than Granite 3.2 8B or MiniCPM4 8B. This advantage likely arises from two factors: first, the llama.cpp framework is highly optimized for the Llama architecture, with kernels tuned to its tensor shapes and memory layouts; and second, the architecture itself is comparatively lean, employing rotary embeddings and streamlined transformer blocks that reduce computational overhead. Thus, the observed speedup reflects both implementation bias and architectural efficiency.

Architectural choices further modulate runtime behavior. The mixture-of-experts (MoE) model OLMoE-1B-7B constitutes a notable outlier, achieving inference times comparable to 3–4B dense models despite its nominal parameter scale. This result suggests that even for smaller models, MoE can greatly boost inference speeds without sacrificing accuracy.

In summary, the findings demonstrate four main points: (i) inference time scales predictably with input length across all models, (ii) speed advantages of smaller models diminish below the 1B range, (iii) hardware constraints and architectural design substantially affect throughput beyond what parameter counts alone would predict, and (iv) framework-level and architecture-specific optimizations, as seen with the Llama models, can produce consistent gains even within the same parameter class. These considerations are critical when selecting models for tasks dominated by either short or long documents, as they emphasize the need to balance model size, accuracy, and hardware efficiency in practical deployments.

10. Accuracy and Computational Cost Balance

We present a visualization that captures the trade-off between models' normalized accuracy on sentiment analysis tasks and their throughput when applied at scale (Figure 8). This representation allows for a direct comparison of how well models balance predictive performance against computational efficiency. To further clarify these trade-offs, we construct a Pareto Front (Figure 9), which highlights the most efficient model at each performance tier and makes explicit the frontier beyond which improvements in one dimension necessarily entail sacrifices in the other. In doing so, we illustrate how different architectures occupy distinct regions of the accuracy–throughput spectrum, offering insights into the scenarios where each model is most appropriately deployed.

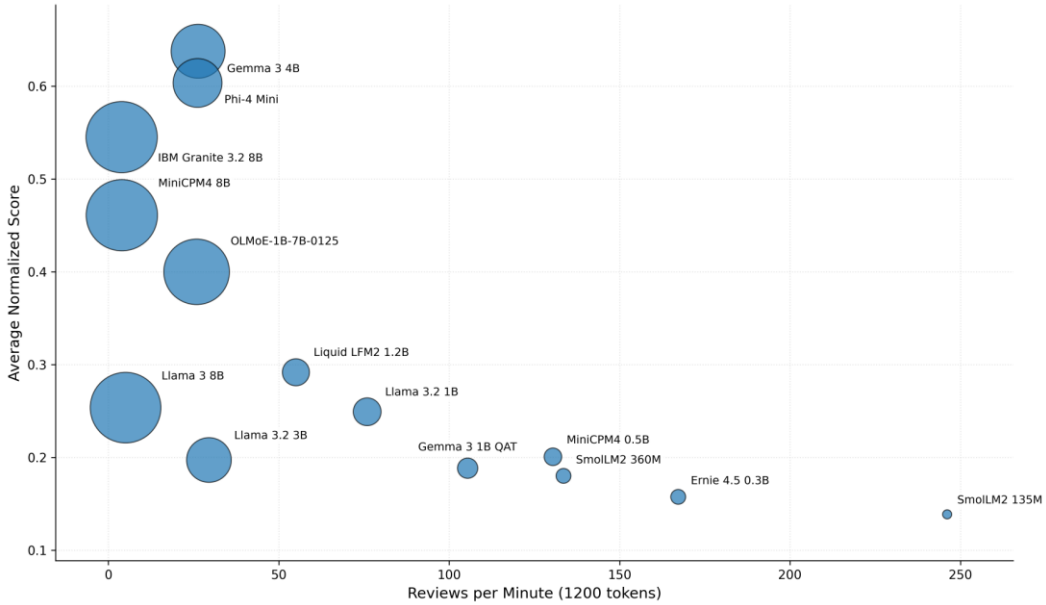

Fig. 8. Normalized Accuracy Score with no penalization vs Reviews per Minute for reviews at 1200 token length. Size of the bubbles correspond to the memory footprints of the models

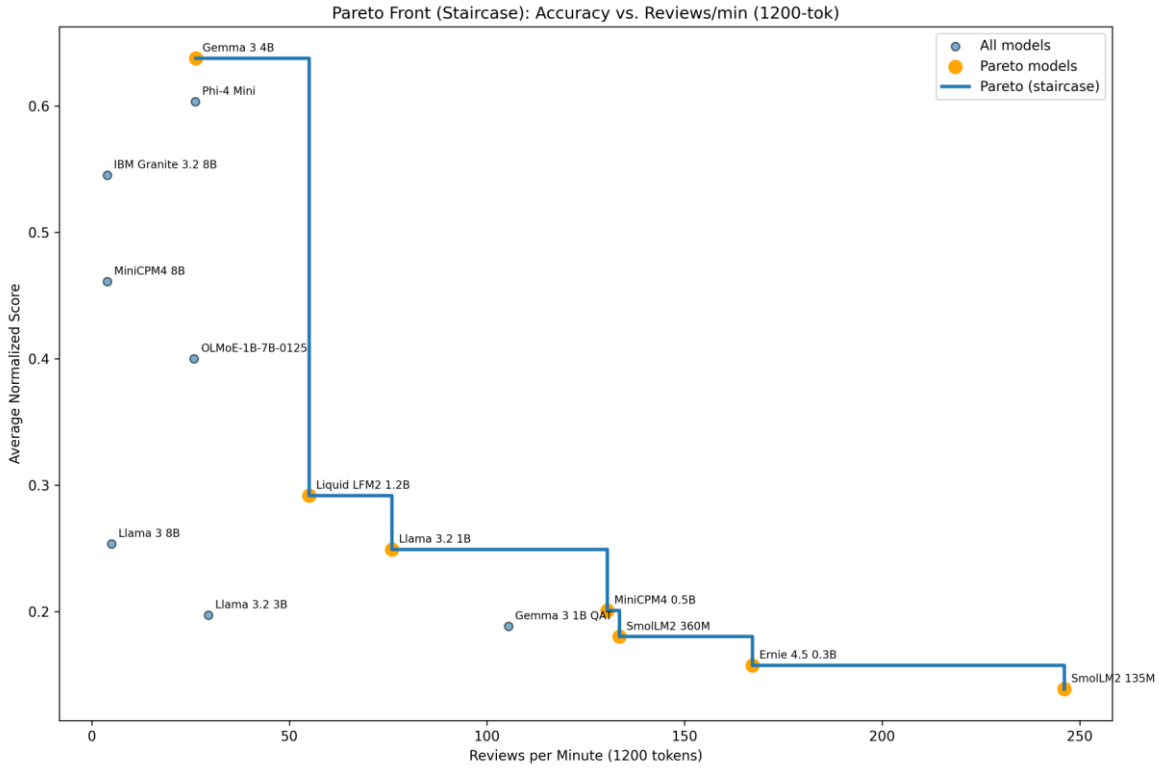

Fig. 9. Pareto Front of the best model at every latency target for zero-shot Sentiment Analysis

Across our zero-shot evaluations, mid-sized 3–4B models: especially Gemma 3 4B and Phi-4 Mini outperform the 7–8B group on our normalized accuracy and in the overall speed–accuracy trade-off. In practice, they deliver accuracy higher than their larger counterparts, while maintaining substantially lower latency and memory pressure, which makes them the most practical default choices for zero-shot sentiment analysis deployments with limited memory budget.

For applications where throughput and latency dominate and some loss in accuracy is acceptable, Liquid LFM2 (1.2B) and Llama 3.2 1B provide markedly faster inference and remain viable in pipelines tolerant of occasional misclassifications. These models are well suited to high-volume screening or interactive settings that must respond within tight time budgets.

When the workflow includes fine-tuning, extremely compact models such as MiniCPM4 0.5B and ERNIE 4.5 0.3B become compelling due to their speed and low memory footprint, enabling rapid iteration and frequent domain refreshes. In scenarios where a model is already fine-tuned for its deployment domain and only needs to execute prompt-based classification at scale, SmolLM2 135M offers the fastest prompt processing among the models we tested.

We also constructed Pareto Fronts for input lengths of 30 and 150 tokens (omitted from the main text). At 150 tokens, Microsoft Phi-4 Mini and OLMoE-1B-7B-0125 join the Pareto frontier; however, their end-to-end prompt ingestion speeds are not materially faster than Gemma 3 4B, so the 3–4B "sweet spot" remains intact unless the deployment context specifically favors those architectures.

11. Discussion

The results of this study highlight the complex balance between accuracy, efficiency, and usability when deploying compact large language models for sentiment analysis. While it is often assumed that larger models will invariably yield better performance, our findings demonstrate that this relationship is not straightforward. In particular, mid-sized models in the 3–4B parameter range, most notably Gemma 3 4B and Microsoft Phi-4 Mini emerged as the most practical defaults. They consistently outperformed some 7–8B models in normalized accuracy while offering substantially lower latency and memory consumption. This finding challenges the prevailing tendency to treat 7–8B models as the minimum viable baseline for serious sentiment analysis tasks.

By contrast, sub-1B models were unable to deliver reliable results in a zero-shot setting. Although they offered impressive speed and memory efficiency, their predictions frequently collapsed to a subset of labels or defaulted to majority classes. This behavior was especially pronounced on fine-grained datasets such as SST-5 and Emotions, where smaller models struggled to capture subtle distinctions between closely related categories. Nevertheless, these compact models retain potential in scenarios where fine-tuning is possible or where the deployment context prioritizes throughput over classification quality. In such cases, their lightweight nature enables frequent retraining, rapid iteration, and deployment on resource-limited hardware.

At the high end of the parameter spectrum, 7–8B models did achieve strong results on binary and domain-specific sentiment tasks, but their relative advantage was diminished once normalized against dataset-specific baselines. Furthermore, their high memory footprint and latency at longer input lengths raise practical concerns for real-world use. The observed slowdown was not solely a function of parameter count but also reflected hardware bottlenecks, particularly when GPU memory capacity was exceeded and inference spilled into shared system RAM. This underscores the importance of considering system-level constraints alongside raw model capabilities when selecting architectures for deployment.

Our evaluation also revealed instances where framework-level optimizations strongly influenced performance. For example, the Llama family consistently processed long inputs faster than

comparably sized alternatives, an advantage likely attributable to llama.cpp's architecture-specific tuning. Similarly, the OLMoE-1B-7B model benefited from its mixture-of-experts design, delivering inference speeds more typical of mid-sized dense models. These cases illustrate how runtime behavior cannot be attributed solely to model size or architecture but also depends on the surrounding software and inference stack.

From a practical standpoint, the trade-offs identified here point toward differentiated deployment strategies. For accuracy-sensitive applications that must still operate efficiently, mid-sized models appear to offer the best balance. For extremely high-throughput or edge-constrained scenarios, compact sub-1B models may still be viable if paired with task-specific fine-tuning. Conversely, when absolute accuracy is paramount and hardware resources are not a limiting factor, 7–8B models remain a robust, albeit less efficient, choice.

Finally, several limitations of this work must be acknowledged. First, all experiments were conducted in a zero-shot setting on English-language datasets; the extent to which these results generalize to multilingual contexts or fine-tuned workflows remains an open question. Second, the experiments were performed on a single hardware platform (Radeon 890M via Vulkan), and different backends or GPU architectures may yield different relative results. Third, only instruction-tuned non-reasoning variants were considered, leaving open the possibility that reasoning-augmented models may alter the trade-offs observed here. Addressing these limitations presents opportunities for future research, particularly in exploring cross-lingual sentiment tasks, incorporating fine-tuning strategies, and extending the evaluation to diverse hardware environments.

12. Conclusion

This paper presented a systematic benchmark of compact large language models, ranging from 135M to 8B parameters, on diverse sentiment analysis datasets. By normalizing results across tasks with different levels of difficulty and class imbalance, we demonstrated that performance does not scale linearly with model size. Instead, models in the 3–4B parameter range, particularly Gemma 3 4B and Microsoft Phi-4 Mini delivered the most favorable balance of accuracy, latency, and memory usage. These mid-sized architectures consistently outperformed or matched the results of larger 7–8B models while avoiding the prohibitive computational overhead that hinders deployment in practical settings.

At the same time, our analysis showed that sub-1B models remain limited in zero-shot conditions, often defaulting to trivial predictions. Nevertheless, their speed and minimal memory footprint make them promising candidates for scenarios involving fine-tuning or extremely resource-constrained deployment. Conversely, 7–8B models retain their value in cases where absolute accuracy is prioritized above efficiency, though their heavy computational requirements limit their accessibility for many real-world applications.

Taken together, these findings suggest that the "sweet spot" for zero-shot sentiment analysis lies in the mid-sized model category, offering a pragmatic compromise between usabil-ity and reliability. Future work should extend this benchmark to multilingual datasets, incorporate fine-tuning and reasoning-augmented variants, and evaluate performance across diverse hardware backends. Such efforts will be essential for understanding how compact models can be most effectively integrated into sentiment analysis pipelines, particularly in domains where efficiency, accessibility, and privacy are critical.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems, pp. 5998–6008, 2017.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., "Language models are few-shot learners," Advances in neural information processing systems, vol. 33, pp. 1877–1901, 2020.

[3] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., "Gpt-4 technical report," arXiv preprint arXiv:2303.08774, 2023.

[4] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," in Proceedings of the AAAI conference on artificial intelligence, vol. 34, pp. 13693–13696, 2020.

[5] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient fine-tuning of quantized LLMs," Advances in neural information processing systems, vol. 36, pp. 10088–10115, 2023.

[6] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in Proceedings of the ACL-02 conference on Empirical methods in natural language processing, pp. 79–86, Association for Computational Linguistics, 2002.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), pp. 4171–4186, 2019.

[8] 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.

[9] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W.and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," Journal of machine learning research, vol. 21, no. 140, pp. 1–67, 2020.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in Proceedings of Workshop at ICLR, 2013.

[11] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, 2014.

[12] 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.

[13] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in International Conference on Learning Representations (ICLR), 2020.

[14] N. Thanh Pham, T. Kieu, D.-M. Nguyen, S. H. Xuan, N. Duong-Trung, and D. Le- Phuoc, "Slm-bench: A comprehensive benchmark of small language models on environmental impacts–extended version," arXiv e-prints, pp. arXiv–2508, 2025.

[15] Z.-H. Tan, Z.-C. Zhao, H.-Y. Shi, X.-Y. Zhang, P. Tan, Y. Yu, and Z.-H. Zhou, "Learnware of language models: Specialized small language models can do big," arXiv preprint arXiv:2505.13425, 2025.

[16] P. Lepagnol, T. Gerald, S. Ghannay, C. Servan, and S. Rosset, "Small language models are good too: An empirical study of zero-shot classification," arXiv preprint arXiv:2404.11122, 2024.

[17] L. Couto Seller, Í. Sanz Torres, A. Vogel-Fernández, C. González Carballo, P. M. Sánchez Sánchez, A. Carruana Martín, and E. d. M. Ambite, "Evaluating compact llms for zero-shot iberian language tasks on end-user devices," arXiv e-prints, pp. arXiv–2504, 2025.

[18] F. Koto, T. Beck, Z. Talat, I. Gurevych, and T. Baldwin, "Zero-shot sentiment analysis in low-resource languages using a multilingual sentiment lexicon," arXiv preprint arXiv:2402.02113, 2024.

[19] Y. Liu, X. Zhu, Z. Shen, Y. Liu, M. Li, Y. Chen, B. John, Z. Ma, T. Hu, Z. Li, et al., "Do large language models possess sensitive to sentiment?," arXiv preprint arXiv:2409.02370, 2024.

[20] T. Fawcett, "An introduction to ROC analysis," Pattern recognition letters, vol. 27, no. 8, pp. 861–874, 2006.

[21] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pp. 142– 150, 2011.

[22] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in Proceedings of the 2013 conference on empirical methods in natural language processing, pp. 1631–1642, 2013.

[23] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multitask benchmark and analysis platform for natural language understanding," arXiv preprint arXiv:1804.07461, 2018.

[24] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, "Good debt or bad debt: Detecting semantic orientations in economic texts," Journal of the Association for Information Science and Technology, vol. 65, no. 4, pp. 782–796, 2014.

[25] Crowdflower, "Twitter us airline sentiment." https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment , 2015.

[26] E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen, "Carer: Contextualized affect representations for emotion recognition," in Proceedings of the 2018 conference on empirical methods in natural language processing, pp. 3687–3697, 2018.

[27] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Predicting the type and target of offensive posts in social media," arXiv preprint arXiv:1902.09666, 2019.

[28] M. Saeidi, G. Bouchard, M. Liakata, and S. Riedel, "Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods," arXiv preprint arXiv:1610.03771, 2016.

[29] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," Advances in neural information processing systems, vol. 32, 2019.

[30] G. Nkhata, S. Gauch, U. Anjum, and J. Zhan, "Fine-tuning bert with bidirectional LSTM for fine-grained movie reviews sentiment analysis," arXiv preprint arXiv:2502.20682, 2025.

[31] "Glue benchmark leaderboard." https://gluebenchmark.com/leaderboard/

[32] V. M., "Text-classification-financial-phrase-bank." https://github.com/vrunm/Text-Classification-Financial-Phrase-Bank , 2023.

[33] M. M. Rahman, A. I. Shiplu, Y. Watanobe, and M. A. Alam, "Roberta-bilstm: A context-aware hybrid model for sentiment analysis," IEEE Transactions on Emerging Topics in Computational Intelligence, 2025.

[34] J. Yan, P. Pu, and L. Jiang, "Emotion-rgc net: A novel approach for emotion recognition in social media using roberta and graph neural networks," Plos one, vol. 20, no. 3, p. e0318524, 2025.

[35] Z.Wu, H. Zheng, J.Wang, W. Su, and J. Fong, "Bnu-hkbu uic nlp team 2 at semeval- 2019 task 6: Detecting offensive language using bert model," in Proceedings of the 13th international workshop on semantic evaluation, pp. 551–555, 2019.

[36] Z. Wu and D. C. Ong, "Context-guided BERT for targeted aspect-based sentiment analysis," in Proceedings of the AAAI conference on artificial intelligence, vol. 35, pp. 14094–14102, 2021.

[37] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integerarithmetic- only inference," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2704–2713, 2018.

[38] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al., "Qwen3 technical report," arXiv preprint arXiv:2505.09388, 2025.

[39] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al., "The llama 3 herd of models," arXiv e-prints, pp. arXiv–2407, 2024.

[40] I. Granite Team, "Granite 3.0 language models," URL: https://github. com/ibmgranite/ granite-3.0-language-models, 2024.

[41] M. Team, C. Xiao, Y. Li, X. Han, Y. Bai, J. Cai, H. Chen, W. Chen, X. Cong, G. Cui, et al., "Minicpm4: Ultra-efficient LLMs on end devices," arXiv preprint arXiv:2506.07900, 2025.

[42] N. Muennighoff, L. Soldaini, D. Groeneveld, K. Lo, J. Morrison, S. Min, W. Shi, P. Walsh, O. Tafjord, N. Lambert, et al., "Olmoe: Open mixture-of-experts language models," arXiv preprint arXiv:2409.02060, 2024.

[43] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," Neural computation, vol. 3, no. 1, pp. 79–87, 1991.

[44] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin,T. Matejovicova, A. Ramé, M. Rivière, et al., "Gemma 3 technical report," arXiv preprint arXiv:2503.19786, 2025.

[45] A. Abouelenin, A. Ashfaq, A. Atkinson, H. Awadalla, N. Bach, J. Bao, A. Benhaim, M. Cai, V. Chaudhary, C. Chen, et al., "Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras," arXiv preprint arXiv:2503.01743, 2025.

[46] Liquid AI, "Introducing lfm2: The fastest on-device foundation models on the market,"

https://www.liquid.ai/blog/liquid-foundation-models-v2-our-second-series-of-generative-ai-models

[47] L. B. Allal, A. Lozhkov, E. Bakouch, G. M. Blázquez, G. Penedo, L. Tunstall, A. Marafioti, H. Kydlíček, A. P. Lajarín, V. Srivastav, et al., "Smollm2: When smol goes big–data-centric training of a small language model," arXiv preprint arXiv:2502.02737, 2025.

[48] B. .-E. Team, "Ernie 4.5 technical report." https://ernie.baidu.com/blog/publication/ERNIE_Technical_Report.pdf , 2025.

[49] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," arXiv preprint arXiv:2010.16061, 2020.

[50] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," Biochimica et Biophysica Acta (BBA)-Protein Structure, vol. 405, no. 2, pp. 442–451, 1975.

[51] J. Cohen, "A coefficient of agreement for nominal scales," Educational and psychological measurement, vol. 20, no. 1, pp. 37–46, 1960.

[52] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (RMSE) in assessing average model performance," Climate research, vol. 30, no. 1, pp. 79–82, 2005.

[53] A. Webson and E. Pavlick, "Do prompt-based models really understand the meaning of their prompts?," in Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: Human language technologies, pp. 2300–2344, 2022.

[54] F. Jiang, Z. Xu, L. Niu, B. Y. Lin, and R. Poovendran, "Chatbug: A common vulnerability of aligned LLMs induced by chat templates," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, pp. 27347–27355, 2025.

[55] K. Lyu, H. Zhao, X. Gu, D. Yu, A. Goyal, and S. Arora, "Keeping LLMs aligned after fine-tuning: The crucial role of prompt templates," Advances in Neural Information Processing Systems, vol. 37, pp. 118603–118631, 2024.

[56] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, et al., "Holistic evaluation of language models," arXiv preprint arXiv:2211.09110, 2022.

[57] Z. Zhao, E.Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," in International conference on machine learning, pp. 12697–12706, PMLR, 2021.

[58] Y. Fei, Y. Hou, Z. Chen, and A. Bosselut, "Mitigating label biases for in-context learning," arXiv preprint arXiv:2305.19148, 2023.