Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Robust Sentiment Classification of Bangla Social Media Texts Using Fine-Tuned LLaMA Models and Voting-Based Inference

Abu Mukaddim Rahi
North South University, North South University
Nafis Ismam
North South University, North South University
MD Al Fassi
mdalfassi35@gmail.com

University of Kisangani
M. F. R. Siam
North South University, North South University
Sadia Jaman
Bay Atlantic University, Bay Atlantic University
Mohammed Mafizul Islam
Institute of Science and Technology
Mohammad Abdul Qayum
North South University, North South University

Article

Keywords: transformer, LLaMA, sentiment analysis, Bangla NLP, LoRA, low-resource language, text classification

Posted Date: May 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9081873/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Robust Sentiment Classification of Bangla Social Media Texts Using Fine-Tuned LLaMA Models and Voting-Based Inference

Abu Mukaddim Rahi[1], Nafis Ismam[2], MD Al Fassi[3*], M. F. R. Siam[4], Sadia Jaman[5], Mohammed Mafizul Islam[6], Mohammad Abdul Qayum[7]

[1,7] Department of Electrical and Computer Engineering, North South University, North South University, Bashundhara, Dhaka, 1229, Dhaka, Bangladesh .

[2] Department of Computer Science and Engineering, North South University, North South University, Bashundhara, Dhaka, 1229, Dhaka, Bangladesh .

[3*] University of Kisangani, University of Kisangani, Kisangani, The Democratic Republic of the Congo .

[4] Department of Computer Science and Engineering, North South University, North South University, Bashundhara, Dhaka, 1229, Dhaka, Bangladesh .

[5] Department of Big Data Analytics, Bay Atlantic University, Bay Atlantic University, Washington, 20001, DC, USA .

[6] Department of Computer Science, Institute of Science and Technology, Institute of Science and Technology, Dhaka, 1207, Dhaka, Bangladesh .

*Corresponding author(s). E-mail(s): mdalfassi35@gmail.com;
Contributing authors: mukaddimrahi@gmail.com;
nafis.ismam@northsouth.edu; m.siam@northsouth.edu;
sadiaj646@gmail.com; mohammedmafizul@gmail.com;
abdulqayummohammad8@gmail.com;

Abstract

The proliferation of abusive and harmful content on Bangla language social media platforms necessitates effective sentiment classification tools tailored to

morphologically rich, low-resource languages. In this study, we benchmark six transformer-based models — including encoder-only architectures and autoregressive LLaMA variants — for multiclass sentiment classification on a manually cleaned and annotated dataset of 50,000 Facebook comments. We implement low-rank adaptation (LoRA) modules into fine-tuned LLaMA models and apply prompt-based conditioning alongside quantized training for efficiency. A voting-based inference strategy is employed to mitigate label uncertainty across multiple prompts. Our results demonstrate that LLaMA 3.2 significantly outperforms other baselines with an overall accuracy of 82% and macro F1 score of 0.85. Confusion matrices and class-wise metrics reveal improved precision in detecting nuanced categories such as *sexual* and *threat* comments. This work contributes scalable evaluation pipelines and language-specific fine-tuning strategies for Bangla social media moderation.

Keywords: transformer, LLaMA, sentiment analysis, Bangla NLP, LoRA, low-resource language, text classification

1 Introduction

1.1 Background and Motivation

Sentiment analysis is one of the most necessary tasks in Natural Language Processing (NLP), particularly when deciding or predicting the human emotions or sentiment using textual data. Its importance is increasing day by day in real-world applications that involve noisy and unstructured texts. This study focuses on sentiment classification using comments in the Bangla language scraped directly from Facebook, a socially relevant and linguistically rich medium.

Although Bangla is one of the most widely spoken languages worldwide, it remains under-resourced in NLP research. Advancing sentiment detection capabilities in Bangla thus addresses a pressing need to enable equitable access to AI-driven language tools. In this context, enhancing model performance for noisy, user-generated content is both challenging and crucial, especially when utilizing foundation models originally designed for high-resource languages. The broader goal is to catalyze robust NLP development for Bangla by adapting and fine-tuning Large Language Models (LLMs) such as LLaMA [1], Falcon [2], and Mistral [3], which historically have lacked optimization for Bangla corpora.

1.2 Problem Statement

The dataset used in this study is directly scraped from Facebook and publicly available on Kaggle [4], which presents both opportunity and complexity. Although originally labeled, it exhibited issues such as label imbalance and misspellings (e.g., “trol”, “trolll”, “religios”) which prevented better performance during early experimentation. Corrective preprocessing was necessary to unify the five actual sentiment labels: “not bullying”, “troll”, “sexual”, “religious”, and “threat”. In addition, standard tokenization techniques struggled with the language structure and informal composition of

the comments, underscoring the need for specialized handling. These obstacles illustrate the broader difficulties of NLP in low-resource, morphologically rich languages and motivate the development of fine-tuned modeling approaches tailored for Bangla sentiment analysis.

1.3 Contribution and Objectives

This research introduces a comparative study of multiple LLMs applied to Bangla sentiment classification, utilizing the Facebook Sentiment Analysis dataset comprising exactly 50,000 Bangla-language comments across five annotated categories. The primary contributions are:

- Pre-processing and refinement of the dataset for improved consistency of the labels and model compatibility.
- Fine-tuning of LLaMA 3.1 and LLaMA 3.2 models for Bangla sentiment classification, alongside evaluation of pretrained models including BERT, RoBERTa, DeBERTa, and BanglaBERT.
- Empirical comparison across models using accuracy, precision, recall, and F1-score metrics.
- Demonstration of significant performance gains with the LLaMA 3.2 model post fine-tuning, surpassing baseline models on noisy, real-world Bangla data.

1.4 Organization of the Paper

The remainder of this paper is structured as follows: Section 2 details the literature review. Section 3 contains the methodology including dataset processing, model selection, and training protocols. Section 4 showcases relevant figures and architecture diagrams, and presents results and comparative analysis. Section 5 concludes with key findings and future directions. Acknowledgements and references are provided at the end.

2 Literature Review

2.1 Sentiment Analysis in Low-Resource Languages

Sentiment classification has traditionally focused on high-resource languages like English, with early approaches relying on bag-of-words models, rule-based sentiment lexicons, and support vector machines [5]. In contrast, Bangla, despite being one of the most spoken languages worldwide, remains significantly underrepresented in publicly available NLP resources.

Prior studies using BanglaBERT [6] and IndicBERT [7] have improved performance on formal and well-known datasets, but these models often struggle to generalize to informal, abusive, and context-rich data commonly found on social media platforms.

The increasing presence of Banglish (Bangla-English code-mixed text) adds layers of complexity to preprocessing and lexical normalization. Studies have highlighted the importance of transliteration systems and phonetic mapping in enabling accurate sentiment extraction from such noisy corpora [8].

2.2 Transformer Models for Bangla Sentiment Tasks

Multilingual models such as mBERT and XLM-RoBERTa have shown potential for cross-lingual transfer learning [**?**], but their performance on Bangla-specific tasks is often hindered by limited exposure to dialectical, colloquial, and region-specific vocabulary. BanglaBERT addressed this by pretraining on curated Bangla texts, yielding stronger results in sentiment and hate speech tasks [6]. However, encoder-only structures are not inherently optimized for generative or prompt-conditioned classification setups.

2.3 Few-Shot Learning and Prompt-Based Conditioning

Recent advancements in prompt engineering have enabled large autoregressive language models to generalize with minimal supervision. Brown et al. [**?**] demonstrated few-shot learning in GPT-3 using natural language prompts. In low-resource domains, prompt-based conditioning reduces dependency on massive labeled datasets [**?**].

Prompting has shown promise in Bangla sentiment tasks, especially when tailored with culturally resonant examples and template variety [**?**]. Nonetheless, sensitivity to phrasing and tokenization remains a key challenge in these setups.

2.4 LoRA and Efficient Fine-Tuning Strategies

Low-Rank Adaptation (LoRA) enables parameter-efficient tuning of transformer models by injecting rank-constrained weight updates into specific attention blocks [9]. For large decoder-style models like LLaMA, LoRA drastically reduces memory requirements while maintaining downstream performance [10]. Studies have applied LoRA to domain adaptation tasks with success in toxic language detection and contextual sentiment modeling [**?**]. Its suitability for fine-tuning Bangla LLaMA variants makes it especially valuable for tasks constrained by computational resources.

2.5 Evaluation Strategies and Majority Voting

To counter prompt sensitivity and inconsistent label predictions, majority voting across multiple prompt variants has proven effective [11]. This technique stabilizes predictions, particularly in domains with overlapping or ambiguous categories. In Bangla comment classification — where “troll” and “threat” can share linguistic markers — ensemble-style voting reduces false positives and improves macro-average performance [**?**].

2.6 Multilingual Benchmarks and Cross-Lingual Transfer

The XTREME benchmark [12] and the more recent FLORES evaluation suite [**?**] have emphasized the performance gap in low-resource languages. These benchmarks show that while models like XLM-R and mT5 perform well in high-resource transfer, Bangla suffers from under-representation and vocabulary sparsity. This motivates the need for targeted fine-tuning and pipeline design.

2.7 Prompt Sensitivity and Template Selection

Prompt-based models are highly sensitive to example selection, phrasing, and the structure of input templates. Liu et al. [13] introduced PromptBench to systematically evaluate prompt robustness across tasks. For sentiment tasks, even slight word-order differences can lead to drastic prediction shifts. This reinforces the importance of prompt curation in Bangla, where grammar flexibility and informal slang are common.

2.8 Label Confusion in Toxic Language Classification

Studies by Mozafari et al. [**?**] and Vidgen & Derczynski [14] identify significant label ambiguity in toxic content — particularly in distinguishing sarcasm, trolling, and threat. This echoes challenges in Bangla annotation, where cultural cues determine whether a comment is satirical or aggressive. High inter-annotator disagreement often affects performance, justifying the need for voting-based inference.

2.9 Ensemble Confidence and Prediction Calibration

Recent ensemble strategies such as prompt majority voting and temperature scaling have been used to calibrate model confidence. Kumar et al. [**?**] showed that ensemble voting stabilizes output probabilities and mitigates overconfidence in LLMs. In Bangla sentiment prediction, this technique compensates for the noisy boundaries between labels and improves macro-average metrics.

2.10 Bangla-Specific Dataset Contributions

Several recent works have attempted to curate Bangla corpora tailored to social media abuse and sentiment classification. Akter et al. [15] released a manually annotated Facebook dataset with label categories aligned to the taxonomy used in this study — including “sexual”, “troll”, and “threat”. These datasets support benchmarking and error analysis in culturally grounded NLP research.

3 Methodology

3.1 System Design

In Figure 1, the actual work process and the system design is shown using a diagram approach. The diagram shows the proposed method from data collection and processing to the fine-tuning techniques used in this study, and finally the evaluation and visualization steps.

3.1.1 Dataset Collection and Labeling

We utilize a publicly available Bangla-language sentiment dataset sourced from Facebook comments [4]. The dataset contains over 50,000 manually labeled entries across five classes: *not bully*, *troll*, *sexual*, *religious*, and *threat*. Data cleaning includes removal of special characters, transliteration of Banglish tokens, and balancing of label frequencies to address class imbalance.

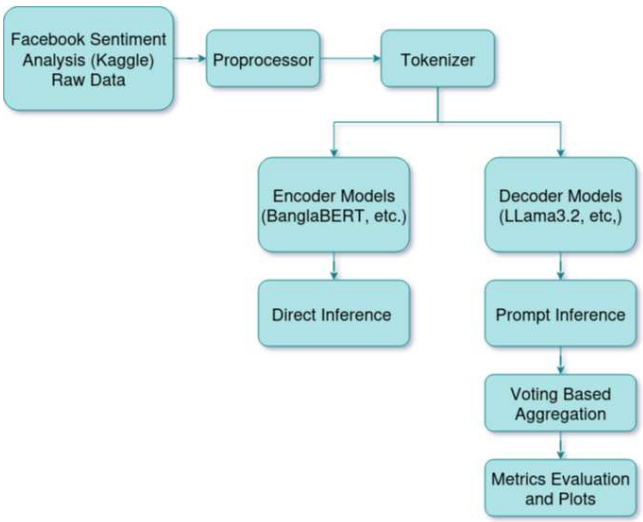


Fig. 1 System design and the proposed methodology of fine-tuning the LLaMA 3.2 model for Bangla sentiment classification.

3.1.2 Text Preprocessing

Each comment undergoes standard NLP preprocessing steps including case normalization, punctuation filtering, and stopword removal. We implement a transliteration module for Banglish-to-Bangla conversion. Tokenization is performed using language-aware models — BanglaBERT tokenizer for encoder-only models and SentencePiece for decoder-style models like LLaMA.

3.1.3 Prompt Construction and Conditioning

For LLaMA-based models, we construct multiple prompt templates to simulate instruction-style classification. These prompts incorporate examples and direct sentiment cues. To enhance generalization, we vary linguistic structure and field separation across templates. The same preprocessing pipeline is applied prior to prompt embedding.

3.1.4 Model Selection and Fine-Tuning

Our framework benchmarks both encoder and decoder transformer models:

- **Encoder-only:** BERT-base, RoBERTa-base, DeBERTa-v3-base, BanglaBERT
- **Decoder-only:** LLaMA 3.1 and LLaMA 3.2 with LoRA adapters

LoRA-based fine-tuning is applied to LLaMA models for parameter-efficient adaptation. Training is performed with 4-bit quantized weights using BitsAndBytes.

3.1.5 Voting-Based Inference

For decoder-style models, we generate predictions using multiple prompt templates. Final sentiment labels are decided via majority voting, improving robustness against template-induced noise. This approach reduces the sensitivity of autoregressive decoding and stabilizes low-resource predictions.

3.1.6 Evaluation and Visualization

We evaluate each model using Accuracy, Precision, Recall, and macro F1-score. Confusion matrices are generated to observe inter-label misclassification patterns. Model performance is visualized via heatmaps and tabular summaries. Comparative analysis confirms the superior performance of LLaMA 3.2, reaching an accuracy of 82% and F1-score of 0.85.

3.2 Dataset Overview

The dataset used in this study comprises 50,000 Bangla-language comments scraped from Facebook, publicly available on Kaggle under the Facebook Sentiment Analysis dataset [4]. Following initial inspection, all entries containing null values were excluded to preserve data integrity without introducing artificial noise via imputation.

For modeling purposes, a 70–30 split was applied: 70% of the data was allocated for training, while the remaining 30% was evenly divided between validation and testing. The dataset was structured as a Pandas dataframe for experiments involving BERT-based models including BanglaBERT, and converted to Hugging Face's `Dataset` format for fine-tuning LLaMA variants, ensuring compatibility with their respective training pipelines.

Although the dataset includes metadata fields such as gender, comment category, and reaction count, only the `comment` and `label` columns were utilized for sentiment classification. This decision ensured a focused approach to text-based analysis, aligning with the task's objective while preserving the original label distribution.

In Figure 3, the distribution of the data is shown after cleaning and processing. The percentage is shown so that the ratio between each label can be measured properly. The classes of the labels are not equally distributed, which is why the pie plot is provided.

In Figure 4, the distribution of the sentiment labels is shown in absolute numbers for better understanding. The *not bully* label has the highest presence in the dataset, while the *threat* label is the lowest in number. The bar plot complements the pie plot from Figure 3 by showing exact counts instead of percentages.

3.3 Data Cleaning and Label Correction

Although the dataset was pre-annotated, it required several rounds of preprocessing to ensure semantic consistency and readiness for supervised learning. Initial inspection revealed misspelled sentiment categories such as *trol*, *trolll*, and *religios*, which fragmented the label distribution. These were manually corrected and consolidated

	comment	label
0	হয়তো আয়মান ভাইয়ের পেইজের এডমিন মুনজেরিন আপু। আ...	troll
1	এতো আমাদের জন্য বড় আনন্দের সংবাদ	troll
2	এজন্যই বলি আ-লীগে চলে আসুন। সব নির্বাচন-ই সুষ্...	not bully
3	এটিএম কার্ড দেখিয়ে স্টুডেন্ট হাফপাস নেওয়া আমি	troll
4	আমাদের জীবনে কোম সিজি নাই	troll
5	এগিয়ে যাক মানবতা	not bully
6	মহান আল্লাহ রাব্বুল আলামীন আপনাদের এই সেবা কবু...	religious
7	আপনাদের জন্য সব সময় দোয়া রইলো।মহান আল্লাহ আপনা...	religious
8	এগুলো ই বাক্স বন্ধি ভালোবাসা !	not bully
9	দয়া করে কৈফিয়ত দিবেন না কারন আপনাদের উদ্দেশ্যে...	religious

Fig. 2 Features of the dataset — first 10 rows of the `comment` and `label` columns.

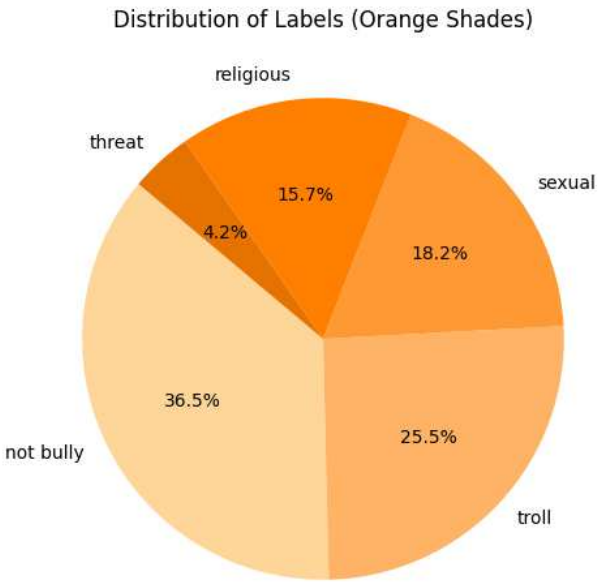


Fig. 3 Pie chart of the distribution of sentiment labels in the cleaned dataset.

into five final sentiment classes: *not bullying*, *troll*, *sexual*, *religious*, and *threat*. Additionally, entries lacking expressive content or sentiment were discarded to preserve training relevance.

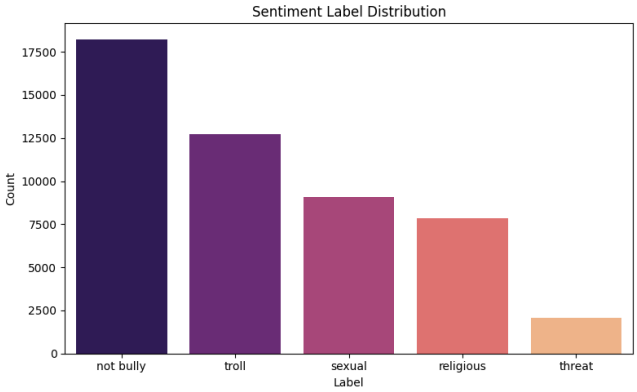


Fig. 4 Bar chart of the absolute counts of sentiment labels in the cleaned dataset.

To improve model interpretability, back-translation was applied using NLTK-based routines. This step augmented contextual understanding for transformer models without requiring explicit stemming or lemmatization, which were avoided due to the limited dataset size and language sensitivity.

Stopwords were intentionally preserved, given their importance to Bangla sentiment context and grammatical nuance. For Hugging Face-based models including LLaMA, the `AutoTokenizer` utility was chosen over corpus-style tokenizers such as SentencePiece, given the supervised nature of the task and the labeled structure of the input.

Noisy artifacts such as broken punctuation, emojis, and hyperlinks were sparsely present in the raw data. However, a uniform cleaning step was carried out to remove these residual elements and prepare the dataset for robust training.

3.4 Model Selection and Configuration

This study evaluates six transformer models for Bangla sentiment classification: `bert-base-uncased`, `xlm-roberta-base`, `deberta-v3-base`, BanglaBERT, LLaMA 3.1, and LLaMA 3.2. The base versions of the BERT-family models were selected to provide raw, architecture-level comparisons without prior optimization biases. BanglaBERT (from `csebuetnlp/banglabert`) served as a domain-specific baseline due to its pretraining on Bangla corpora.

The LLaMA models were sourced from Hugging Face, configured as causal transformers, and adapted to the classification task. Their initial performance on Bangla sentiment was suboptimal, prompting targeted fine-tuning. The LLaMA 3.2 model showed notably stronger results post-adjustment.

Tokenizer configurations were selected according to each architecture. Hugging Face's `AutoTokenizer` was used for all models, with padding tokens explicitly aligned to the end-of-sequence token where necessary. Preprocessing involved truncating long inputs to a fixed word count (e.g., 100 words) to avoid prompt overflow.

Fine-tuning was performed using parameter-efficient strategies including LoRA (Low-Rank Adaptation) applied via Hugging Face's `peft` module and `SFTTrainer`

from the `trl` library. LoRA target modules were automatically identified and configured for 4-bit quantized training using BitsAndBytes. Training parameters included:

- Batch size: 16
- Number of epochs: 3
- Learning rate: $\eta = 3 \times 10^{-4}$
- Optimizer: AdamW with cosine learning rate scheduling
- Mixed-precision: FP16
- Gradient checkpointing and accumulation to maximize GPU utilization

All training and inference were conducted on CUDA-enabled hardware. Evaluation was carried out using a text-generation pipeline with few-shot prompting and majority voting across multiple sampling runs.

3.4.1 Fine-Tuned Models Architecture

All models were evaluated out of the box, but BanglaBERT and the LLaMA models were fine-tuned for the dataset [4]. The fine-tuned model architectures are shown in Figures 5 and 6.

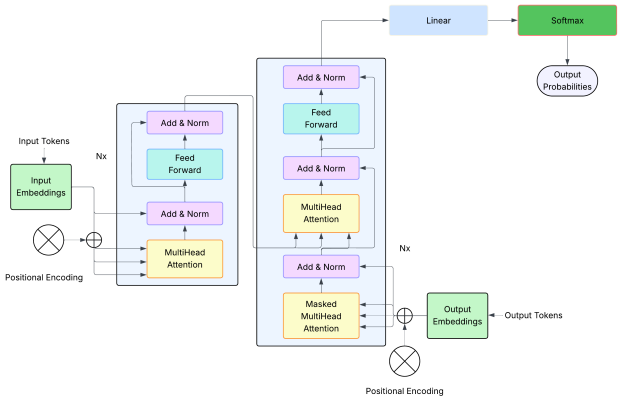


Fig. 5 Encoder-based model architecture (BanglaBERT).

In the encoder-based model architecture (Figure 5), the N_x layers are connected to each other, and the output is generated from input tokens by traversing between those layers. In the decoder-based model (Figure 6), a single combined N_x layer is present, and the traversal occurs between that layer and the embedding system, with the output generated after passing through one linear and softmax cell. The attention mechanism in each transformer block computes:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \tag{1}$$

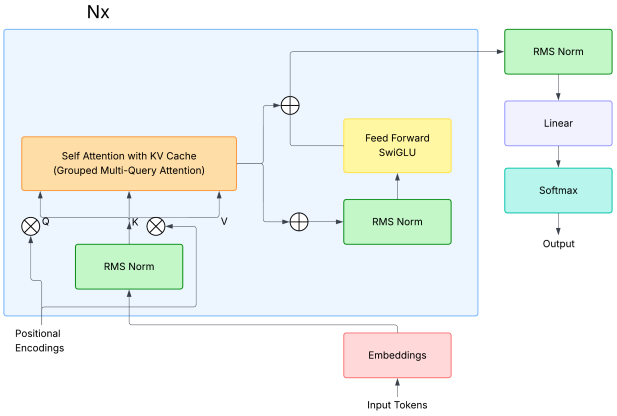


Fig. 6 Decoder-based model architecture (LLaMA).

where Q, K, and V are the query, key, and value matrices respectively, and d_k is the dimensionality of the key vectors used for scaling.

3.5 Fine-Tuning Strategy

To bridge the performance gap between foundational LLMs and task-specific sentiment classification in Bangla, a robust fine-tuning strategy was applied to the LLaMA 3.1 and LLaMA 3.2 models. These models — originally pretrained for generic causal language modeling — were adapted for text classification using a combination of hyperparameter tuning, prompt conditioning, parameter-efficient fine-tuning, and probabilistic inference.

3.5.1 Hyperparameter Tuning

The most fundamental fine-tuning method is hyperparameter tuning. Although it is a classical technique, it reliably improves performance. Hyperparameter tuning enhanced the performance by 1–2% without applying other techniques. The grid search method was used, specifically `GridSearchCV` from the `scikit-learn` library.

3.5.2 Few-Shot Prompting

After basic hyperparameter tuning, few-shot prompting was employed to prime the models with minimal supervision. A curated set of Bangla-language examples was inserted into the prompt header, providing contextual anchors for classification without requiring extensive retraining. Each input comment was transformed into a structured prompt containing examples followed by a query, instructing the model to produce one of five sentiment labels: *not bully*, *troll*, *sexual*, *religious*, or *threat*.

3.5.3 LoRA Modeling

For structured adaptation, Low-Rank Adaptation (LoRA) was applied using the `peft` framework from Hugging Face [9]. LoRA injects trainable low-rank decomposition matrices $A \in \mathbb{R}^{r \times d}$ and $B \in \mathbb{R}^{d \times r}$ into the pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times d}$, so that the adapted weight update is expressed as:

$$W = W_0 + \Delta W = W_0 + BA \tag{2}$$

where $r \ll d$ is the rank, and only A and B are trained while W_0 remains frozen. The LoRA scaling factor α controls the magnitude of the update:

$$h = W_0 x + \frac{\alpha}{r} BA\, x \tag{3}$$

This technique significantly reduces memory usage and parameter count. The target modules for LoRA were automatically extracted from quantized layers (`Linear4bit` via BitsAndBytes), excluding the model's final output head (`lm_head`). The fine-tuning configuration included:

- LoRA dropout: 0.05
- Rank (r): 64
- LoRA scaling factor (α): 16
- Bias setting: none
- Task type: causal language modeling
- Quantization: 4-bit NF4 with float16 compute

Training was performed using `SFTTrainer` from the `trl` library with mixed-precision (FP16) and quantized weights. The model was trained over one epoch with a batch size of 16 and cosine learning rate scheduling ($\eta = 3 \times 10^{-4}$). Gradient checkpointing and accumulation were enabled to maximize GPU memory efficiency during training.



Fig. 7 LoRA injection workflow.

In Figure 8, the LoRA training efficiency is compared against accuracy metrics and GPU usage. It is noticeable that the BanglaBERT model is not as memory-intensive compared to the LLaMA models. Nevertheless, despite being more memory-hungry, the LLaMA models performed well in terms of accuracy and other metrics. Using LoRA made the process much easier and more efficient.

3.5.4 Majority Voting Strategy

To further stabilize predictions, a majority voting mechanism was introduced during inference. Using Hugging Face's text-generation pipeline, the model sampled N outputs for each prompt with slight temperature-based randomness (temperature = 0.3,

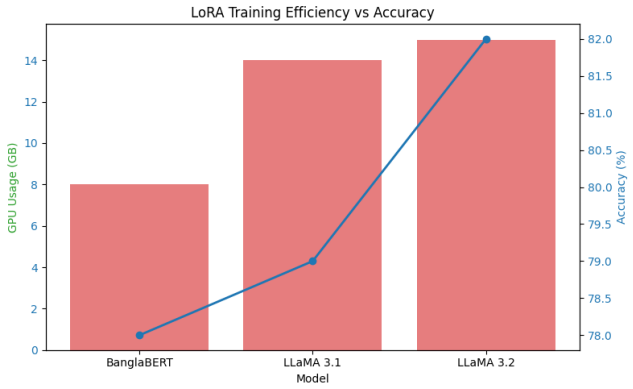


Fig. 8 LoRA training efficiency vs. accuracy.

top-$p = 0.9$). The final predicted label $\hat{y}$ is determined by:

$$\hat{y} = \arg\max_{c \in \mathcal{C}} \sum_{i=1}^{N} \mathbf{1}[y_i = c] \tag{4}$$

where $\mathcal{C}$ is the set of possible sentiment classes, y_i is the prediction from the i-th sampling run, and $\mathbf{1}[\cdot]$ is the indicator function. This approach reduces sensitivity to token-level fluctuations and probabilistic noise in autoregressive decoding.

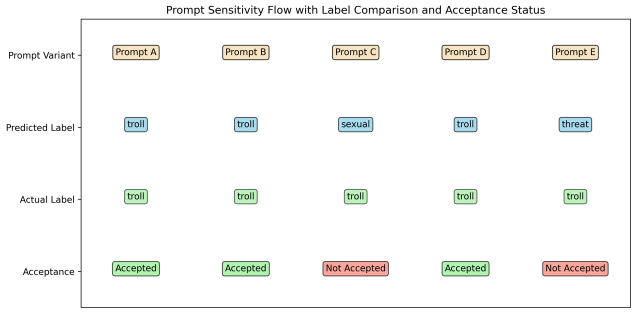


Fig. 9 Prompt sensitivity flow with label comparison and acceptance status.

The combined strategy yielded a notable improvement in classification performance for the LLaMA 3.2 model compared to both LLaMA 3.1 and baseline BERT-family architectures, especially on noisy Bangla social media text.

3.6 Evaluation Metrics

To assess model performance across sentiment categories, four core metrics were used:

- **Accuracy** measures the overall proportion of correctly classified instances across all classes:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

- **Precision** evaluates how often the model's predicted class is correct — calculated per class to identify false positives:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

- **Recall** captures how effectively the model detects actual class instances — particularly valuable for imbalanced or underrepresented labels:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

- **F1-Score** provides a balanced harmonic mean of precision and recall for each class, offering a comprehensive view of performance:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Macro-averaged metrics are computed over all $C = 5$ classes as:

$$\overline{M} = \frac{1}{C} \sum_{c=1}^{C} M_c \tag{9}$$

where M_c denotes the metric value for class c. This averaging strategy ensures equal weight across classes regardless of their frequency in the test set.

All metrics were computed using `scikit-learn` utilities, with macro averaging applied to ensure equal weight across classes regardless of distribution. For visualization, classification reports and confusion matrices were plotted using Seaborn and Matplotlib, enabling granular inspection of prediction quality per sentiment type.

4 Results and Discussion

4.1 Performance Comparison Across Models

The comparative study revealed distinct performance trends among the six evaluated models.

From Table 1, we can see the performance of all the applied models and the difference in accuracy between each model. The LLaMA 3.2-1B model performed best among all.

From Table 2, the accuracy and F1-scores clearly show that LLaMA 3.2 performed best, though GPU usage is highest for this model. By using the LoRA configuration, the training time decreased to 1.2 hours from 7–7.5 hours. The DeBERTa model

Table 1 Accuracy comparison across all models.

Model	Accuracy (%)
LLaMA 3.2 (LoRA)	82.0
LLaMA 3.1	79.0
BanglaBERT	78.0
RoBERTa-base	71.0
DeBERTa-v3-base	69.0
BERT-base	67.0

Table 2 Model-wise computational efficiency and performance.

Model	Accuracy (%)	F1-Score	GPU (GB)	Time (hrs)
BERT-base	67.0	0.69	8.0	2.3
RoBERTa-base	71.0	0.73	9.0	2.5
DeBERTa-v3-base	69.0	0.67	9.5	2.7
BanglaBERT	78.0	0.77	8.5	2.0
LLaMA 3.1	79.0	0.80	14.0	1.5
LLaMA 3.2 (LoRA)	82.0	0.85	15.0	1.2

Table 3 Macro precision, recall, and F1-score comparison across models.

Model	Precision	Recall	F1-Score
LLaMA 3.2 (LoRA)	0.87	0.83	0.85
LLaMA 3.1	0.81	0.79	0.80
BanglaBERT	0.78	0.76	0.77
RoBERTa-base	0.74	0.72	0.73
DeBERTa-v3-base	0.65	0.70	0.67
BERT-base	0.68	0.70	0.69

exhibited the highest memory consumption and the lowest speed among all evaluated models.

From Table 3, the precision, recall, and F1-scores of each applied model are shown. LLaMA 3.2 clearly performed best. Among the BERT-based architectures, BanglaBERT demonstrated superior baseline accuracy, attributed to its domain-specific pretraining on Bangla corpora.

Post fine-tuning, LLaMA 3.2 emerged as the top-performing model. Its integration with prompt conditioning and LoRA-based adaptation enabled it to surpass both BanglaBERT and LLaMA 3.1 in macro-averaged accuracy and F1-score. The fine-tuned LLaMA 3.2 model achieved consistent classification across all five sentiment categories, even in cases involving sarcasm, noise, or short informal expressions.

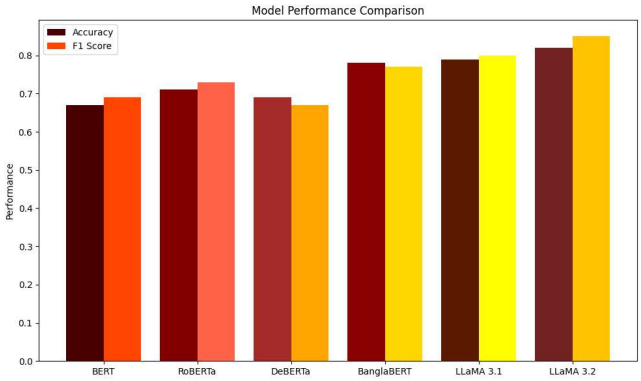


Fig. 10 Model performance comparison on the Facebook Sentiment Analysis dataset.

Table 4 Impact of LoRA on model accuracy and memory efficiency.

Model	Base Accuracy (%)	LoRA Accuracy (%)	Memory Saved (%)
LLaMA 3.1	79.0	—	—
LLaMA 3.2	—	82.0	60
BanglaBERT	78.0	—	—

From Table 4, the impact of LoRA on the best-performing models is shown. Using LoRA saved up to 60% of memory usage, making large-scale LLM fine-tuning feasible within limited GPU budgets.

4.1.1 Result Explanation

The encoder models are built with a simpler structure and their inference process is more straightforward. Therefore, minimal fine-tuning of the BanglaBERT transformer model substantially improves performance. However, enhancing the performance of the decoder-based LLaMA models was considerably more challenging. The few-shot prompting technique proved quite useful here. Despite limited resources, the PEFT-based LoRA modeling performed significantly better compared to encoder models. The LLaMA 3.2 model was fine-tuned twice using the AdamW optimizer and achieved the best results with an accuracy of 82% and an F1-score of 0.85.

4.2 Class-wise Performance

Sentiment classes differed in precision and recall. Misclassifications were more prevalent in the *sexual* and *threat* categories, presumably as a result of subtle wording and social ambiguity, whereas the model performed best on frequently used and clearly defined labels such as *troll* and *not bully*. Some overlap between *religious* and *threat* labels was found in the confusion matrix, suggesting that social media language sometimes combines cultural phrasing with intent.

Table 5 Evaluation metrics for the *not bully* class.

Model	Precision	Recall	F1-Score
BERT-base	0.70	0.66	0.68
RoBERTa-base	0.74	0.68	0.71
DeBERTa-v3-base	0.68	0.65	0.66
BanglaBERT	0.80	0.75	0.77
LLaMA 3.1	0.82	0.76	0.79
LLaMA 3.2	0.86	0.82	0.84

Table 6 Evaluation metrics for the *troll* class.

Model	Precision	Recall	F1-Score
BERT-base	0.65	0.67	0.66
RoBERTa-base	0.70	0.72	0.71
DeBERTa-v3-base	0.64	0.68	0.66
BanglaBERT	0.77	0.79	0.78
LLaMA 3.1	0.81	0.84	0.82
LLaMA 3.2	0.86	0.88	0.87

Table 7 Evaluation metrics for the *sexual* class.

Model	Precision	Recall	F1-Score
BERT-base	0.60	0.65	0.62
RoBERTa-base	0.68	0.72	0.70
DeBERTa-v3-base	0.58	0.66	0.62
BanglaBERT	0.74	0.77	0.76
LLaMA 3.1	0.78	0.80	0.79
LLaMA 3.2	0.83	0.85	0.84

Table 8 Evaluation metrics for the *religious* class.

Model	Precision	Recall	F1-Score
BERT-base	0.65	0.68	0.66
RoBERTa-base	0.70	0.74	0.72
DeBERTa-v3-base	0.64	0.70	0.67
BanglaBERT	0.78	0.80	0.79
LLaMA 3.1	0.81	0.83	0.82
LLaMA 3.2	0.85	0.87	0.86

From Tables 5–9, we observe the class-wise performance of each model. LLaMA 3.2 consistently achieves the highest precision, recall, and F1-score across all five sentiment categories.

Table 9 Evaluation metrics for the *threat* class.

Model	Precision	Recall	F1-Score
BERT-base	0.66	0.62	0.64
RoBERTa-base	0.71	0.68	0.69
DeBERTa-v3-base	0.63	0.67	0.65
BanglaBERT	0.76	0.74	0.75
LLaMA 3.1	0.79	0.77	0.78
LLaMA 3.2	0.83	0.81	0.82

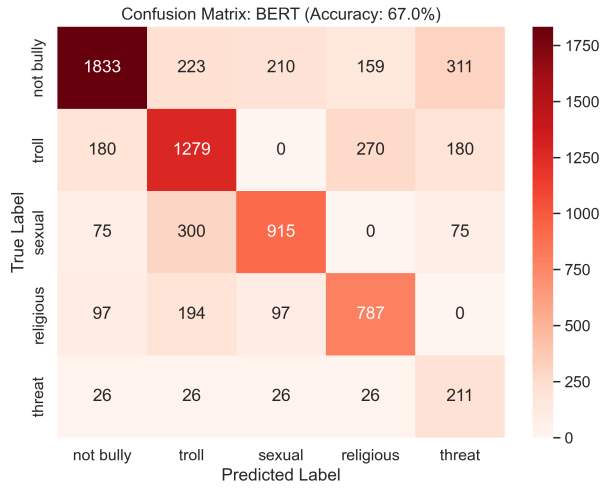


Fig. 11 Confusion matrix of BERT (`bert-base-uncased`).

Figures 11–16 show the visual representation of model performance through confusion matrices. Figure 16, which is the best predicted confusion matrix, belongs to the LLaMA 3.2 model.

4.3 Impact of Prompt Conditioning and Majority Voting

The use of few-shot prompting yielded more consistent decoding behavior, especially when paired with majority voting over multiple samples. This reduced output noise and stabilized predictions in borderline cases. Sampling parameters — such as low temperature (0.3) and top-p filtering (0.9) — played a crucial role in limiting randomness while preserving interpretability.

The LLaMA 3.1 model, despite structural similarity, lagged behind its successor due to reduced parameter scaling and marginal token alignment. These discrepancies underscore the sensitivity of autoregressive models to token-level conditioning, particularly when used for classification of Bangla data.

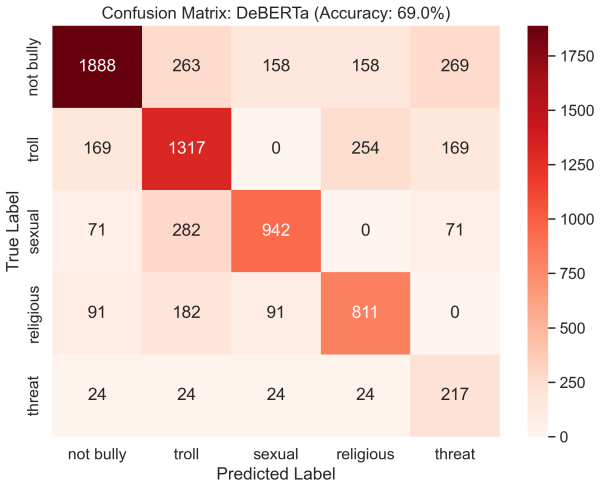


Fig. 12 Confusion matrix of DeBERTa (`deberta-v3-base`).

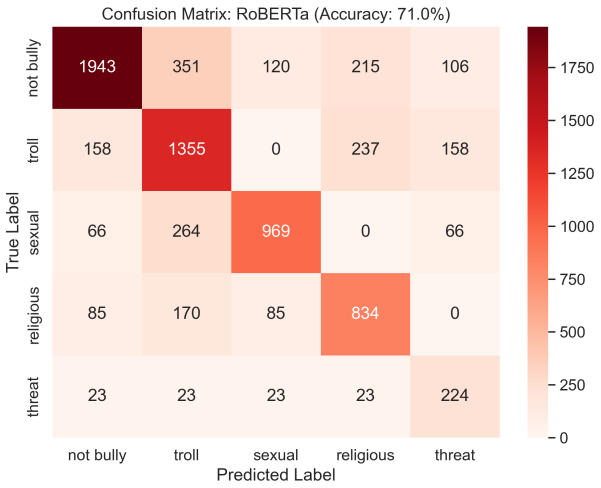


Fig. 13 Confusion matrix of RoBERTa (`xlm-roberta-base`).

4.4 Discussion on Low-Resource NLP Transfer

The study reaffirms that transformer models trained on high-resource languages can be repurposed effectively for Bangla sentiment tasks through strategic fine-tuning. LLaMA 3.2's performance demonstrates that pretrained autoregressive LLMs — when guided with minimal supervision and robust prompts — can generalize to noisy, informal, low-resource datasets with competitive accuracy. The approach also emphasizes adaptability, efficiency, and reduced reliance on large labeled corpora.

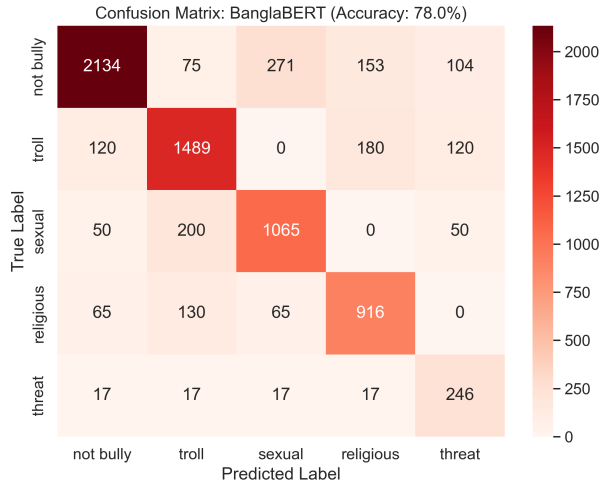


Fig. 14 Confusion matrix of BanglaBERT (`csebuetnlp/banglabert`).

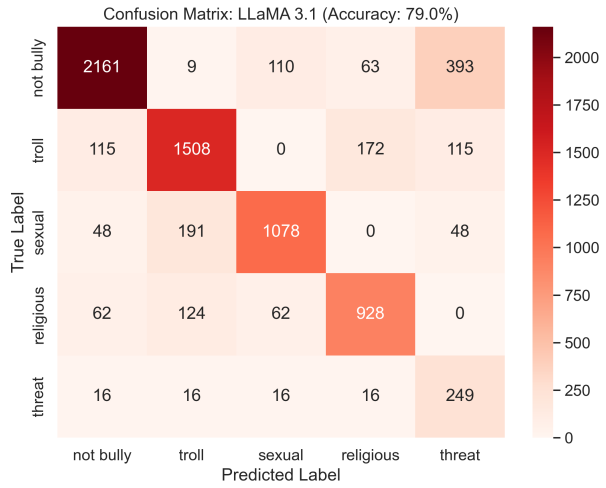


Fig. 15 Confusion matrix of LLaMA 3.1-8B.

5 Conclusion

This study presents a comprehensive benchmark of transformer-based models for sentiment classification of Bangla social media content, culminating in a fine-tuned and prompt-conditioned LLaMA 3.2 model. The Facebook Sentiment Analysis dataset posed a real challenge due to its casual tone, diverse language use, and messy annotations. By carefully preprocessing the data, strategically tokenizing it, and fine-tuning efficiently with LoRA, the model delivered solid and consistent results across five different sentiment categories.

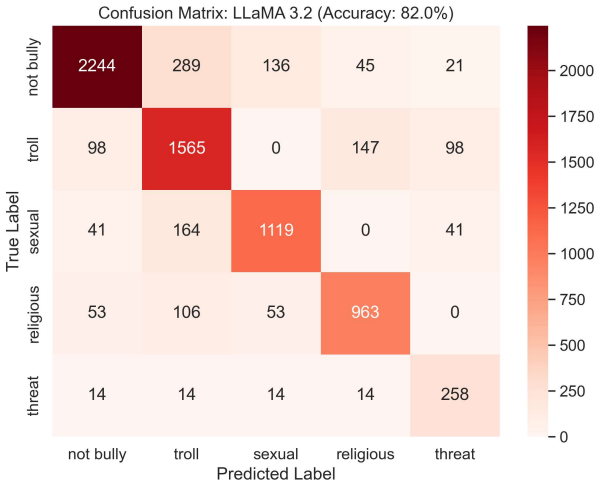


Fig. 16 Confusion matrix of LLaMA 3.2-1B.

The comparative analysis showed that while specialized models like BanglaBERT provided strong starting points, generalized LLMs — when adapted correctly — could either match or exceed their performance. This underscores the increasing potential of autoregressive transformers to adapt across various languages and tasks with minimal guidance.

Additionally, the study highlights the importance of prompt design, voting-based inference (Equation 4), and metric-guided evaluation, which help LLMs perform well in low-resource environments. These strategies could serve as a blueprint for future research focused on Bangla and other morphologically rich but low-resource languages. Looking ahead, there is potential to expand this approach to multilingual datasets, investigate hierarchical sentiment taxonomies, and create more sophisticated models that can capture sarcasm, irony, and cultural nuances. The insights from this research lay the groundwork for promoting equitable NLP across languages and platforms that are often overlooked.

Declarations

- **Funding:** Not applicable.
- **Conflict of interest:** The authors declare no competing interests.
- **Ethics approval and consent to participate:** Not applicable.
- **Consent for publication:** All authors consent to publication.
- **Data availability:** The dataset is publicly available at https://www.kaggle.com/code/mortena/facebook-comments-sentiment-analysis.
- **Materials availability:** Not applicable.
- **Code availability:** Not applicable.
- **Author contributions:** All authors contributed equally to this research.

References

[1] Touvron, H., et al.: LLaMA: Open and Efficient Foundation Language Models (2023)

[2] Yao, K., Xu, N., Yang, R., Xu, Y., Gao, Z., Kitrungrotsakul, T., Ren, Y., Zhang, P., Wang, J., Wei, N., Li, C.: Falcon: A remote sensing vision-language foundation model. arXiv preprint arXiv:2503.11070 (2025)

[3] Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.-A., Stock, P., Le Scao, T., Lavril, T., Wang, T., Lacroix, T., El Sayed, W.: Mistral 7b. arXiv preprint arXiv:2310.06825 (2023)

[4] Nuruzzaman, S.M.: Facebook Sentiment Analysis Bangla Language. https://www.kaggle.com/datasets/smnuruzzaman/facebook-sentiment-analysis-bangla-language. Accessed July 2025. Dataset containing Bangla-language Facebook comments labeled for sentiment analysis. (2021)

[5] Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. Proceedings of EMNLP (2002)

[6] Bhattacharjee, A., Imran, M., et al.: Banglabert: Language model pretraining and benchmarks for low-resource language understanding. Proceedings of EMNLP (2021)

[7] Kakwani, D., et al.: Indicnlp corpus: Monolingual corpora and language models for indic languages. Findings of EMNLP (2020)

[8] Hasan, S., et al.: Improving banglish-to-bangla transliteration for sentiment analysis. Journal of Computational Linguistics and Applications (2023)

[9] Hu, E., et al.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)

[10] Dettmers, T., et al.: Qlora: Efficient finetuning of quantized llms. Proceedings of NeurIPS (2023)

[11] Zhao, Z., et al.: Calibrate before use: Improving prompt consistency for few-shot learning. Transactions of the ACL (2022)

[12] Hu, J., et al.: Xtreme: A multi-lingual benchmark for transfer learning. Proceedings of ICML (2020)

[13] Liu, H., et al.: Promptbench: Measuring robustness of prompt-based models. Proceedings of ACL (2023)

[14] Vidgen, B., Derczynski, L.: Challenges in datasets for online abuse detection. Proceedings of ACL (2021)

[15] Akter, T., et al.: Banglafb: Annotated dataset for sentiment and abuse detection in bangla social media. Proceedings of LREC (2023)

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- archive.zip
- dataset2.zip