5. International Conference on Electrical, Computer and Energy Technologies (ICECET 2025)
3-6 July 2025, Paris-France

Benchmarking Large Language Models for Sarcasm Detection in Hindi Romanized Texts

Abu Mukaddim Rahi
Dept. of ECE
North South University
Dhaka, Bangladesh
abu.rahi@northsouth.edu

Nafis Ismam
Dept. of ECE
North South University
Dhaka, Bangladesh
nafis.ismam@northsouth.edu

Maher Ali Rusho*
Senior Scientist
Department of Computational Material & Data Analytics
Mr. R Business Corporation (NGO)
Chennai, Tamil Nadu, India
maher.rusho@colorado.edu

Saber Hossain
Dept. of ECE
North South University
Dhaka, Bangladesh
saber.hossain@northsouth.edu

Jannatul Fardous Maliha
Dept. of ECE
North South University
Dhaka, Bangladesh
fardous.maliha@northsouth.edu

Md Amdad Hossain
Dept. of CSE
Chittagong University of Engineering & Technology
Dhaka, Bangladesh
u1704100@student.cuet.ac.bd

Mithila Arman
Dept. of CSE
BRAC University
Dhaka, Bangladesh
mithila.arman@g.bracu.ac.bd

Md. Khurshid Jahan
Dept. of ECE
North South University
Dhaka, Bangladesh
khurshid.jahan@northsouth.edu

***Abstract*—Sarcasm detection is a critical challenge in natural language processing (NLP) due to its intricate reliance on context, tone, and implicit meaning. Accurately identifying sarcasm is essential for sentiment analysis, opinion mining, and various downstream NLP applications, including social media monitoring, customer feedback analysis, and chat-bot development. With the rise of large language models (LLMs), significant advancements have been made in many NLP tasks, ranging from machine translation to text generation. However, the capability of these models in detecting sarcasm, particularly in Hindi Romanized text a widely used informal writing style among Hindi speakers remains underexplored.**

This study presents a comprehensive comparative analysis of twelve pretrained LLMs, assessing their effectiveness in sarcasm detection using a Hindi Romanized dataset. The evaluation highlights notable performance variations across models, emphasizing the complexities associated with processing Romanized Hindi text. Factors such as transliteration inconsistencies, code mixing, and the lack of explicit contextual markers contribute to the unique challenges faced by these models. Additionally, the findings offer insights into the strengths and limitations of different LLM architectures in handling nuanced language constructs like sarcasm.

By identifying key challenges and model specific performance trends, this research aims to bridge the gap in sarcasm detection for Hindi Romanized text, providing valuable implications for future model development and dataset curation in multilingual NLP.

***Index Terms*—Sarcasm Detection, Large Language Models, Hindi Romanized Text, Natural Language Processing, Comparative Analysis**

I. Introduction

Sarcasm, a linguistic phenomenon where the intended meaning diverges from the literal wording, presents a persistent challenge in natural language processing (NLP). Unlike explicit expressions of sentiment, sarcasm often relies on contextual cues, tone, and cultural familiarity, making it difficult to detect purely from textual data. Failure to correctly identify sarcasm can lead to significant misclassifications in sentiment analysis, opinion mining, and conversational AI applications, ultimately distorting insights derived from user-generated content [1]. Given the increasing reliance on automated systems for tasks like social media monitoring, customer feedback analysis, and misinformation detection, improving sarcasm detection models is essential to ensuring accurate interpretation of textual data.

The recent advancements in large language models (LLMs) have significantly improved the ability of NLP systems to understand and generate human-like text. Models such as BERT, RoBERTa, GPT, and DeBERTa have demonstrated remarkable capabilities in capturing syntactic and semantic nuances across multiple languages. However, despite their success in various NLP tasks, these models still struggle with more complex linguistic constructs such as sarcasm. Unlike sentiment classification, which relies on direct lexical cues, sarcasm often involves implicit meaning, irony, and exaggeration, which are difficult for models to grasp without deeper contextual understanding [2]. Moreover, sarcasm detection

979-8-3315-3559-9/25/$31.00

becomes even more challenging in non-standard text formats, where linguistic rules and orthographic consistency are not well-defined.

Hindi Romanized text, a widely used form of informal digital communication, further complicates sarcasm detection due to its non-standardized nature. Instead of using the native Devanagari script, Hindi speakers often write in the Roman alphabet, particularly on social media, messaging platforms, and online forums. While this form of writing enhances accessibility, it introduces unique linguistic complexities that pose challenges for NLP models:

- Orthographic Variations: The absence of a standardized spelling system leads to multiple representations of the same word, making tokenization and word embedding techniques less effective. For instance, the word "no" in Hindi may appear as "nahi," "nai," or "nhi," complicating text normalization and model training [3].
- Ambiguity: Romanized Hindi words often have multiple phonetic representations, leading to potential misinterpretations. Unlike Devanagari, where diacritics provide clear phonemic distinctions, Romanized Hindi relies on user preferences, making it harder for models to distinguish words that share similar spellings but differ in meaning [4].
- Code-Switching: A significant portion of Hindi Romanized text involves code-switching, where users mix Hindi and English words within a single sentence. This hybrid linguistic structure introduces additional complexity, as models must simultaneously process multiple grammatical frameworks and contextual dependencies [5].

Despite these challenges, there has been limited research on the effectiveness of pretrained LLMs in sarcasm detection for Hindi Romanized text. Given the growing importance of multilingual NLP, it is imperative to analyze how well these models adapt to informal, non-standardized language forms. This study aims to bridge this research gap by conducting a comparative evaluation of twelve pretrained LLMs on a Hindi Romanized sarcasm dataset. Through this analysis, we seek to identify model-specific strengths and weaknesses, assess the impact of linguistic challenges on sarcasm detection accuracy, and provide insights into improving sarcasm recognition for low-resource and code-mixed languages.

II. Related Work

Sarcasm Detection with GPT Models: This study assesses the differences in sarcasm detection between GPT models with and without domain context, testing fine-tuned and zero-shot models of varying sizes [6].

SarcasmBench Evaluation: This research introduces SarcasmBench, a comprehensive benchmark designed to evaluate LLMs on sarcasm understanding, highlighting performance gaps and challenges [2].

Multimodal Sarcasm Detection: This paper applies multimodal sarcasm detection tasks to the generative framework of multimodal large language models, addressing issues of generalization and multimodal feature reliance [7].

Challenges in Romanized Text Processing: This work addresses the challenges of back-transliteration of Romanized Hindi text, focusing on inconsistencies in spelling and phonetic representation [8].

This was worked on Nepali Romanized data for sentiment analysis. Then they compare performance between BERT and RoBERTa model. They have three types of label(positive,negative,neutral). BERT model performed with 79% accuracy. [9]

Khan et al. worked on Roman Urdu Sentiment Analysis Using Transfer Learning. This paper proposed Convolution Neural Network(CNN) with an attention mechanism and improve sentiment classification accuracy. [10]

Fahim et al. worked on Back Translation on Romanized Bangla language. This paper back translated with 42,705 samples of bangla romanized data. Their future goal is to use BanglaTLit-PT model with their pre-training 245,727 samples data. [11]

Irum Naz Sodhar et al. worked on Sentiment analysis of Romanized Sindhi text. This paper comprising 100 Romanized Sindhi sentences. It used python tool to classified its sentiment like neutral,positive and negative. This paper faces challenges due to lack of resources for Sindhi Romanized text. [12]

A Hassan et al. worked on Sentiment Analysis on Bangla and Romanized Bangla Text. This paper worked with deep recurrent models. It took 10,000 samples of data and each one annotated by two native bangla speakers. LSTM model achieve better performance for bangla text which is 78% and 55% for Romanized bangla text. [13]

Hafiz Hassaan Saeed et al. worked on Roman Urdu Toxic Comment Classification. This Romanized comments were collected from social media. This paper were used Roman Urdu Toxic(RUT) dataset which had 72,000 manually labeled comments. Its ensemble approach achieves an F1-score of 86.35%. [14]

III. Dataset

A. Dataset Description

The dataset utilized in this study is sourced from Kaggle. It contains 2,108 entries, Hindi-English Romanized social media posts, with sarcastic and non-sarcastic class balance. The sentences exhibit varying levels of code-mixing and informal spellings, each with three attributes: ID, Tweet, and Label. The labels are binary—'YES' indicating sarcasm and 'NO' indicating the absence of sarcasm. This dataset is mainly the Tweeter data on different aspect. The Tweets are written in Hing-Lish(Hindi+English) data which makes the data romanized and hard to work with. The reason of selecting this data was this data was not so famous so the workings of the data are very few. [15]

B. Data Features

In this dataset, there are three features of the dataset. Which are ID, Tweet and Labels. While performing Sarcasm Detection the work was done by using the Tweet and the Label

feature which actually defines the text from where the models will learn if the text is a sarcastic text or not.

TABLE-I Features of the Dataset

Tweet	Label
takeout burrito shielded from cold as though it were week-old newborn	YES
sight of coworkers' stupid fucking faces endured yet again	YES
TIL that @RayuduAmbati is ek number ka chootiya. #AmbatiRayudu #Cricket	NO
aa gayi khan aur roshan ki chamchi jab se in logo ki kali kartoote batayi hai Kangna ne baukhala gay	NO
"Bhai, tumhari wisdom se sab overlooked details become brilliant."	YES

C. Feature Distribution

It consists of 2,108 rows of the dataset with all unique values.
The distribution of the unique feature labels are shown as a pie-plot.

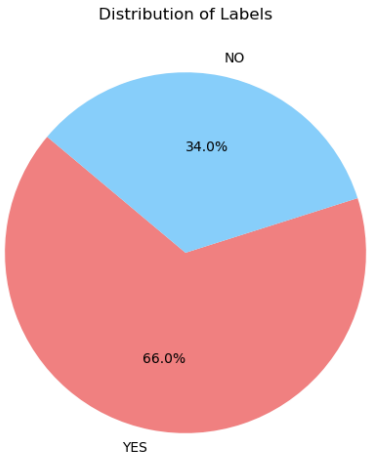

Fig. 1. Pie-Plot of the Feature Distribution of the Dataset.

D. Limitations of the Dataset

The data of the dataset are very noisy and the data consists of Hindi, English and Hinglish data which makes the data more noisy. So the data was processed well before using. The biggest limitation of the dataset is this is a small dataset with only 2108 rows of data. The data were also not equally distributed between the classes of the labels.

IV. Methodology

A. Data Preprocessing

As the dataset were so noisy, to work with the data preprocessing was necessary. As the preprocessing the Tokenization was performed. The text of the data were segmented into smalled tokens which helped the models to recognize the words properly. To handle the noisiness of the dataset the StopWord removal was performed before going to the modelling part. The data were mapped twice to get rid of the formation and oversampling. Then the data were already splitted in the Kaggle dataset. The **train.csv** was to train the model and the **test.csv** was to test the data and validate the data. And, finally the torch formatting was performed in the preprocessing part to make sure it works well with the Torch Library and it becomes compatible with the PyTorch.

B. Proposed Methodology

Need to import necessary libraries so that models can execute properly. After doing pre-processing then pre-processed data is ready for train. Finally performance of all models will measure by evaluation matrix.

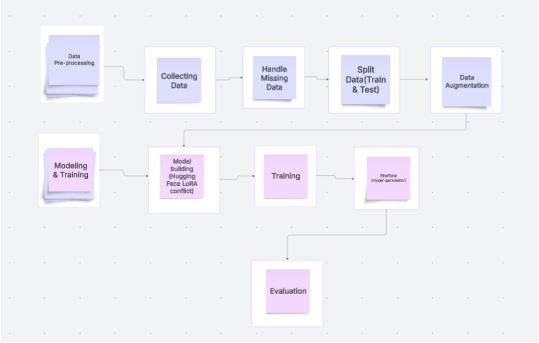

Fig. 2. The Proposed Methodology for Fine-Tuning LLMs for Hindi Romanized Texts

C. Model Selection

Total 12 pretrained LLMs were used in the dataset to perform the Benchmarking the performance.

- BERT
- RoBERTa
- DeBERTa
- ALBERT
- FinBERT
- Mistral
- Phi-4
- Gemma-2b
- Llama3.1
- Llama3.2
- XL-NET
- Electra

D. Training and Fine-Tuning

1) Training: The training for the models were same for the BERT based model like BERT, RoBERTa, DeBERTa, FinBERT, ALBERT was same where the training argument were passed with the specific Learning Rates, Weight Decays, and Epochs. For the models like Mistral, LLama, Gemma the pipeline were called and the model training were performed. XL-net, Electra, Phi-4 models were performing well without using the pipeline so it was normally called with the huggingface training arguments.

2) Fine-Tuning: The Fine-Tuning were performed to enhance the performance of the models on the specific dataset. The fine-tuning techniques like Lora Configurations, BitsAndBytes Configurations were used. For the hyperparameter tuning the GridSearch Algorithm was performed to tune the model with the best parameters.

E. Evaluation Metrics

For evaluating the performance of the LLM models we have used the scikit-learns pretrained Evaluation Metrics like Accuracy, F1-Score, Precision, Recall. For the visualization technique of the performance were visualized by the confusion matrix.

- Accuracy: Accuracy is the representation of correct predictions divided by the total number of predictions, representative of the performance of the model as a whole.

$$\texttt{Accuracy} = \frac{\texttt{TP} + \texttt{TN}}{\texttt{TP} + \texttt{TN} + \texttt{FP} + \texttt{FN}} \tag{1}$$

- Precision: Precision for the positive predictions. It represents the number of relevant instances among the retrieved instances.

$$\texttt{Precision} = \frac{\texttt{TP}}{\texttt{TP} + \texttt{FP}} \tag{2}$$

- Recall: Recall describes how well a model can find all relevant cases in a dataset. This is the percentage of relevant instances of the total relevant instances that were retrieved by the specific model.

$$\texttt{Recall} = \frac{\texttt{TP}}{\texttt{TP} + \texttt{FN}} \tag{3}$$

- F1-Score: The h-mean(harmonic mean) of precision and recall is called F1-Score. It allows to aggregate both metrics and give one value that represents both properties.

$$\texttt{F1-Score} = \frac{2 \times \texttt{Precision} \times \texttt{Recall}}{\texttt{Precision} + \texttt{Recall}} \tag{4}$$

- Confusion Matrix: Tabular representation showing the True Positives, True Negatives, False Positives and False Negatives.

V. Results Evaluation

In this TABLE II, represents comparison of the accuracy scores of the 12 LLMs. The tables shows the performance difference between the LLMs. This study evaluated models using Accuracy, Precision, Recall, and F1-Score. Given the class imbalance and the nuanced nature of sarcasm detection, F1-Score considered the most indicative of model performance. The models benchmarked include BERT, RoBERTa, DeBERTa, ALBERT, FinBERT, XL-Net, Electra, Gemma, Llama-3.1, Llama-3.2, Phi-4 and Mistral.

TABLE II MODEL ACCURACY

Model	Accuracy
BERT	0.99
RoBERTa	0.98
DeBERTa	0.99
ALBERT	0.98
FinBERT	0.99
XL-Net	0.98
Electra	0.98
Gemma	0.60
Llama-3.1	0.61
Llama-3.2	0.40
Phi-4	0.67
Mistral	0.55

TABLE III PERFORMANCE METRICS OF YES LABEL

Model	Precision	Recall	F1 Score
BERT	0.99	0.99	0.99
RoBERTa	0.98	0.97	0.98
DeBERTa	0.99	0.99	0.99
ALBERT	0.98	1.00	0.99
FinBERT	0.99	0.99	0.99
XL-Net	0.99	0.99	0.99
Electra	0.98	0.99	0.99
Gemma	0.70	0.77	0.74
Llama-3.1	0.77	0.62	0.69
Llama-3.2	0.69	0.59	0.64
Phi-4	0.67	1.00	0.80
Mistral	0.75	0.48	0.58

TABLE IV PERFORMANCE METRICS OF NO LABEL

Model	Precision	Recall	F1 Score
BERT	0.98	0.98	0.98
RoBERTa	0.99	0.99	0.99
DeBERTa	0.99	0.97	0.98
ALBERT	0.99	0.96	0.97
FinBERT	0.98	0.98	0.98
XL-Net	0.98	0.97	0.98
Electra	0.98	0.96	0.97
Gemma	0.19	0.14	0.16
Llama-3.1	0.44	0.57	0.49
Llama-3.2	0.32	0.03	0.06
Phi-4	0.00	0.00	0.00
Mistral	0.40	0.68	0.50

In this TABLE III, and IV, shows how the three metrics, Precision, Recall, and F1 Score are performing on two different labels, Yes, and No for various LLMs. From the 12 LLMs the BERT, DeBERTa, and FinBERT has the best results.

In Figure 2, 3, and 4 the performance of the best three performing models are presented as the form of confusion matrix. Where for each sarcastic label Yes and No, the results are shown.

VI. Best Performing Model

From the analysis we got the best performance for RoBERTa model. The RoBERTa model performed very well along with BERT model in this dataset. The RoBERTa model is a BERT-based model but for better multi-lingual tasks.

In Figure 6 the architecture of the BERT model is given. The BERT model was called from the transformers pretrained models. The BERT model follows RNN or Bi-directional RNN system with the single input and some hidden layers and one output layer with the connection of softmax optimization. In this study the best performing 3 models are BERT, DeBERTa, and FinBERT. The BERT performed the best between these

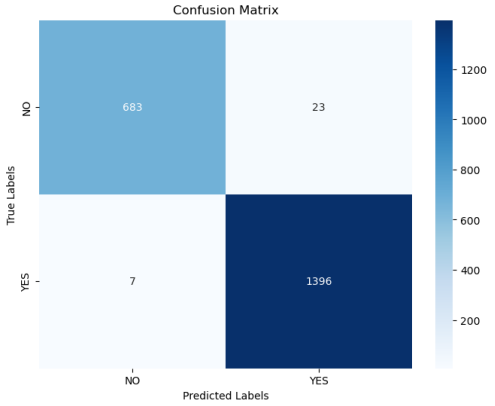

Fig. 3. Confusion Matrix of LLMs (BERT)

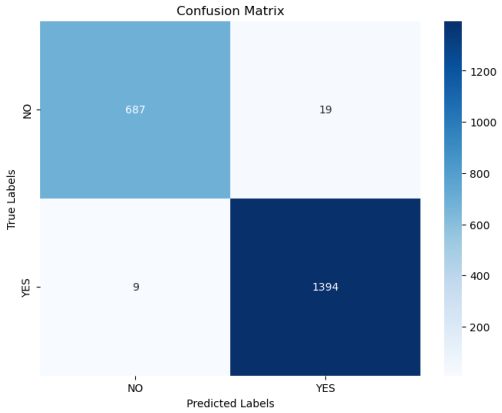

Fig. 4. Confusion Matrix of LLMs (DeBERTa)

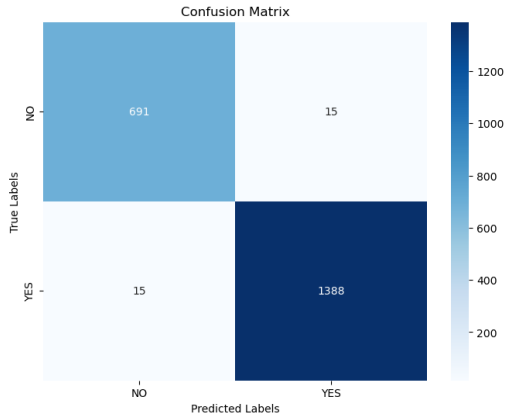

Fig. 5. Confusion Matrix of LLMs (FinBERT)

models. To validate the model predictions, this paper visualize confusion matrices for the top-performing models and plot predicted versus actual sarcasm labels on a sample of test sentences. Code-mixed samples with abrupt Hindi-English switching were often misclassified by RoBERTa but handled better.

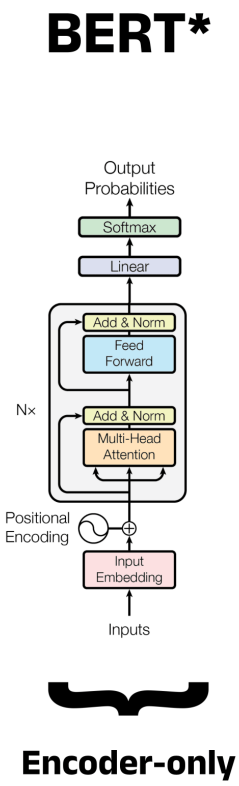

Fig. 6. Architecture of the best performing model BERT

VII. Conclusion and Future Work

In this study, we conducted a comparative analysis of 12 pretrained Large Language Models (LLMs) for sarcasm detection on a Hindi Romanized dataset. Our results demonstrate that transformer-based models such as BERT, DeBERTa, and FinBERT achieved the highest accuracy of 99%, significantly outperforming models like Mistral, Gemma, and LLaMA, which struggled with the complexities of Romanized Hindi. These findings underscore the effectiveness of transformer architectures, particularly those pre-trained on sentiment-heavy corpora, in capturing the implicit and context-dependent nature of sarcasm.

The study highlights the importance of model selection when dealing with sarcasm detection in multilingual and code-mixed settings. While BERT-based models showed excellent performance, other general-purpose LLMs exhibited significant performance drops, likely due to their limited exposure to Romanized Hindi during pretraining. This underscores the necessity of task-specific fine-tuning and domain adaptation when deploying LLMs in low-resource languages. Furthermore, the study emphasizes the challenges associated with processing informal digital text, where non-standard spellings, frequent code-switching, and lack of explicit syntactic markers complicate sarcasm recognition.

Despite the promising results, several challenges and limitations remain. The dataset size (2,108 samples) is relatively

small, which may have contributed to overfitting in high-capacity models. While our analysis provides insights into model performance trends, a larger and more diverse dataset would be necessary to validate these findings comprehensively. Additionally, sarcasm detection remains a linguistically complex problem, especially in Romanized Hindi, where subtle semantic cues and cultural context play a crucial role. Existing LLMs, even those fine-tuned on sentiment-rich corpora, still struggle with highly context-dependent forms of sarcasm, highlighting the need for improved contextual reasoning in NLP models.

Future research should explore several key directions to enhance sarcasm detection in Romanized Hindi and similar low-resource settings. Data augmentation techniques such as back-translation, paraphrasing, and adversarial training could help mitigate data scarcity and improve model generalization. Additionally, transfer learning with Hindi-specific LLMs, particularly those trained on code-mixed and Romanized corpora, could enhance model adaptability to informal linguistic structures. Another promising avenue is multimodal sarcasm detection, integrating textual analysis with prosodic, visual, or contextual cues from social media interactions to improve sarcasm recognition.

Moreover, further research should investigate the interpretability of LLMs in sarcasm detection, analyzing how different architectures handle implicit meaning and whether attention-based mechanisms sufficiently capture irony and context shifts. Expanding the dataset to include a wider range of Romanized Hindi sources—such as movie subtitles, online forums, and conversational AI transcripts—could provide a more robust evaluation benchmark for sarcasm detection models.

This study contributes to the growing field of sarcasm detection in low-resource languages, demonstrating that while LLMs are highly effective, their performance is heavily dependent on training data and language representation. As LLMs continue to evolve, fine-tuning them on linguistically diverse and code-mixed datasets will be crucial for achieving real-world applicability. Addressing the challenges identified in this research will not only improve sarcasm detection but also enhance NLP systems' ability to process informal and multilingual text in diverse digital communication settings.

REFERENCES

[1] TechXplore, "Can large language models detect sarcasm?" *TechXplore*, 2023. [Online]. Available: https://techxplore.com/news/2023-12-large-language-sarcasm.html

[2] Anonymous, "Sarcasmbench: Towards evaluating large language models on sarcasm understanding," *arXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2408.11319

[3] Unknown, "Addressing challenges in automatic language identification of romanized text," *CiteSeerX*. [Online]. Available: https://citeseerx.ist.psu.edu/document?doi=f96237bda9d00ce1dc81897a9b29d05791cf99ff

[4] TridIndia, "Challenges in hindi translation: What makes hindi language more complex," *TridIndia*, 2022. [Online]. Available: https://www.tridindia.com/translation-challenges/challenges-in-hindi-translation/

[5] P. Singhal, "Effective approaches and challenges in hindi-english neural machine translation," *IJHSR*, vol. 4, no. 2, 2022. [Online]. Available: https://terra-docs.s3.us-east-2.amazonaws.com/IJHSR/Articles/volume4-issue2/2022_42_p96_Singhal.pdf

[6] Anonymous, "On sarcasm detection with openai gpt-based models," in *OpenReview*, 2024. [Online]. Available: https://openreview.net/forum?id=wOb0xFwdpr

[7] ——, "Leveraging generative large language models with visual information for multimodal sarcasm detection," in *NAACL*, 2024. [Online]. Available: https://aclanthology.org/2024.naacl-long.97.pdf

[8] ——, "Hindi back transliteration - roman to devanagari," in *Conference on Natural Language Processing*, 2023.

[9] A. Pradhananga and A. K. Sah, "Transformer-based deep learning models for sentiment analysis in romanized nepali: A comparative investigation of bert and roberta," in *Proceedings of the 14th IOE Graduate Conference*, vol. 14. Nepal: Institute of Engineering, Tribhuwan University, December 2023, pp. 296–303. [Online]. Available: https://conference.ioe.edu.np/publications/ioegc14/IOEGC-14-044-C3-1-202.pdf

[10] M. Khan, M. Ali, M. Rauf, D. Javeed, M. Khan, M. Ali, M. Rauf, and D. Javeed, "Roman urdu sentiment analysis using transfer learning," *Applied Sciences*, vol. 12, no. 20, p. 10344, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/20/10344

[11] M. Fahim, F. T. Shifat, F. Haider, D. D. Barua, M. S. U. R. Sourove, M. F. Ishmam, and M. F. A. Bhuiyan, "Banglatlit: A benchmark dataset for back-transliteration of romanized bangla," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024. [Online]. Available: https://aclanthology.org/2024.findings-emnlp.859/

[12] I. N. Sodhar, A. H. Jalbani, A. H. Buller, M. I. Channa, and D. N. Hakro, "Sentiment analysis of romanized sindhi text," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 5877–5883, 2020. [Online]. Available: https://www.researchgate.net/publication/339667632_Sentiment_analysis_of_Romanized_Sindhi_text

[13] A. Hassan, M. R. Amin, A. K. A. Azad, and N. Mohammed, "Sentiment analysis on bangla and romanized bangla text (brbt) using deep recurrent models," *arXiv preprint arXiv:1610.00369*, 2016, accessed: 2025-03-21. [Online]. Available: https://arxiv.org/abs/1610.00369

[14] H. H. Saeed, M. H. Ashraf, F. Kamiran, A. Karim, and T. Calders, "Roman urdu toxic comment classification," *Language Resources and Evaluation*, vol. 55, no. 4, pp. 971–996, 2021. [Online]. Available: https://repository.uantwerpen.be/docman/irua/2fcd7d/a176703.pdf

[15] Divyanshu, "Hackarena theme 2: Multilingual sarcasm detection dataset," 2023, accessed: 2025-03-21. [Online]. Available: https://www.kaggle.com/datasets/divyanshu134/hackarena-theme-2-multilingual-sarcasm-detection