AniSan@NLU of Devanagari Script Languages 2025: Optimizing Language Identification with Ensemble Learning

Anik Mahmud Shanto, Mst. Sanjida Jamal Priya, Mohammad Shamsul Arefin
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
u1904{049, 057}@student.cuet.ac.bd,
sarefin@cuet.ac.bd

Abstract

Identifying languages written in Devanagari script, including Hindi, Marathi, Nepali, Bhojpuri, and Sanskrit, is essential in multilingual contexts but challenging due to the high overlap between these languages. To address this, a shared task on "Devanagari Script Language Identification" has been organized, with a dataset available for subtask A to test language identification models. This paper introduces an ensemble-based approach that combines mBERT, XLM-R, and IndicBERT models through majority voting to improve language identification accuracy across these languages. Our ensemble model has achieved an impressive accuracy of 99.68%, outperforming individual models by capturing a broader range of language features and reducing model biases that often arise from closely related linguistic patterns. Additionally, we have fine-tuned other transformer models as part of a comparative analysis, providing further validation of the ensemble's effectiveness. The results highlight the ensemble model's ability in distinguishing similar languages within the Devanagari script, offering a promising approach for accurate language identification in complex multilingual contexts.

1 Introduction

Effectively processing and comprehending many languages and scripts has become crucial for natural language understanding (NLU) to meet the growing diversity of multilingual content available online. Since Devanagari-scripted languages—such as Hindi, Marathi, Nepali, Bhojpuri, and Sanskrit—are among the most commonly used in South Asia, precise language identification is essential for enabling a wide range of applications, including sentiment analysis, user behavior analysis, and content moderation. In order to meet these needs, CHIPSAL@COLING 2025 (Thapa et al., 2025) has organized a shared task on Natural Language Understanding of Devanagari Script Languages and focused on three main tasks: language identification, hate speech detection, and target identification within hate speech.

Languages written in Devanagari script often share similar sounds, word structures, and sentence patterns. This makes it hard for computers to distinguish between them. The problem is made worse by people often mixing languages, using different regional accents, and using words from dialects. Though several works have been done for Devanagari script language identification using machine learning (Indhuja et al., 2014), deep learning (Sharma and Mithun, 2023) and transformer-based (Thara and Poornachandran, 2021) approaches, the existing works struggle to understand the underlying variances described above.

In the shared task (Sarveswaran et al., 2025), subtask A aims to accurately categorize texts written in Devanagari script into distinct languages. Though almost 2.5 billion people speak these languages, these languages have still been resource-constrained in the NLP research field. Therefore, the organizers have organized this shared task on Devanagari languages to enhance research on these languages. To improve automatic information processing in these languages, further research will help in more sophisticated and accurate identification of these languages. By achieving this, various works like detecting hate speech (Sahoo et al., 2024), determining target for hate speech (Sharma et al., 2024), dialect identification (Das and Bhattacharjee, 2024) etc. can be enhanced towards further improvement.

The primary objective of this task is to detect language from Devanagari scripts. To accomplish this objective, we have developed a number of transformer-based approaches. We have used the provided dataset in the shared task. The key contributions of our research are :

- We have developed an ensemble method that

Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), pages 236–241
January 19, 2025.

leverages the strengths of multiple transformers namely mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020) and IndicBERT (Kakwani et al., 2020).

- We have trained the developed model and evaluated its performance on the provided test set by the organizers.
- We have compared the performance of our developed model with other fine-tuned models.

2 Related Works

Language identification from texts is a popular research topic in natural language processing. Identifying a language from a text involves determining the language or languages present in the written input. Unlike voice, which is processed as a continuous signal, it uses discrete letters, which enable different mathematical techniques to analyze the text (Jauhiainen et al., 2024). An n-gram-based approach using a combination of word search and stop word detection has been proposed in (Pinge et al., 2023). This approach has achieved 95.6% accuracy. In another study, various ML models (Logistic Regression, Decision Tree, Random Forest, and Naive Bayes) have been implemented for language detection. Carpenter (2024) has developed a system using a multinomial naive Bayes algorithm. This system can identify 22 languages with 95% accuracy. Other researchers have also found multinomial naive Bayes effective in language identification task (Sriharsha et al., 2024; Rawat et al., 2023; Menon, 2022). In (R and George, 2023), authors have developed BiLSTM and DCNN-based methods to detect languages like Malayalam, Assamese, Hindi, etc. To identify English Malayalam code-mixed texts, transformer-based models (BERT, CamemBERT, DistilBERT) have been used (Thara and Poornachandran, 2021). This methodology has increased the f1-score by 9% from existing works. Another study has used a transformer-based model to identify code-mixed Kannada texts (Tonja et al., 2022). Finetuning transformers is also an effective way to achieve good performance in various language detection researches (Saifullah et al., 2024; Hossain et al., 2024; Farsi et al., 2024).

3 Dataset and Task

We have participated in subtask A named "Devanagari Script Language Identification". The provided dataset for the task contains 5 languages: Nepali (Thapa et al., 2023; Rauniyar et al., 2023), Marathi (Kulkarni et al., 2021), Sanskrit (Aralikatte et al., 2021), Bhojpuri (Ojha, 2019), and Hindi (Jafri et al., 2023, 2024). Table 1 describes the provided dataset:

Language	Number of Samples		
	Train	Validation	Test
Nepali	12,544	2688	2688
Marathi	11,034	2364	2365
Sanskrit	10,996	2356	2356
Bhojpuri	10,184	2182	2183
Hindi	7664	1643	1642
Combined	**52,422**	**11,233**	**11,234**

Table 1: Distribution of Languages in Datasets (Train, Validation and Test)

Language	Total No. of Words		
	Train	Validation	Test
Nepali	224,033	47,991	48,361
Marathi	273,959	59,134	59,642
Sanskrit	222,568	47,083	46,224
Bhojpuri	292,995	64,460	63,401
Hindi	146,609	32,171	32,254
Combined	**1,160,164**	**250,839**	**249, 882**

Table 2: Word Distribution in Combined Dataset (Train, Validation and Test)

4 System Overview

In our proposed methodology, we have developed an ensemble technique that consists of three transformer-based models. The ensemble technique combines the strengths of multiple transformer-based models to make more accurate predictions. We have ensembled three SOTA transformer-based models.

- **mBERT:** Multilingual BERT or mBERT (Devlin et al., 2019) is a transformer-based model pre-trained on 104 languages, including Devanagari languages. mBERT captures language-agnostic embeddings. Therefore, it has been proven effective in many multilingual tasks.
- **XLM-Roberta:** XLM-RoBERTa or XLM-R (Conneau et al., 2020) is another transformer-based model pre-trained on 100 languages. It captures a wide range of cross-lingual patterns and can handle diverse linguistic syntax.

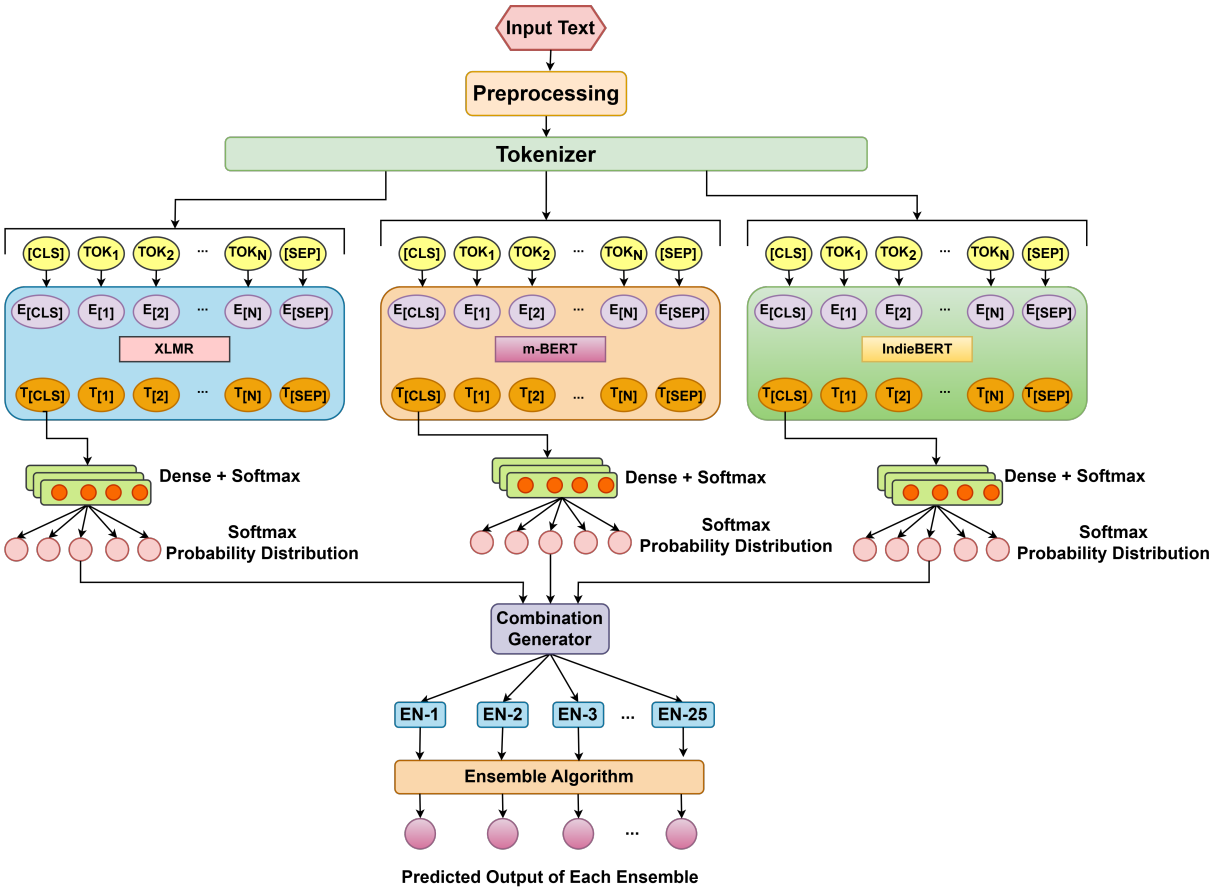


Figure 1: Overview of the methodology

This ability makes it effective for multilingual tasks.

- **IndicBERT:** IndicBERT (Kakwani et al., 2020) is a lightweight transformer model specifically designed for Indic languages. Unlike XLM-R and mBERT, IndicBERT focuses on Indian languages and has been pre-trained on a corpus containing several Devanagari-scripted languages, making it particularly relevant for our task.

For our final predictions, we have used a majority voting technique in combination generation portion of Figure 1 that aggregates the predictions from XLM-R, mBERT, and IndicBERT. Each model independently predicts the language of a given Devanagari-scripted text. The final prediction has been determined based on the majority vote among the three models. This ensemble approach reduces the influence of individual model biases or errors while utilizing the distinct advantages of each model to increase overall performance. Through this approach, our system can more effectively handle the linguistic similarities and complexities within Devanagari-scripted languages, enhancing the precision of language identification in multilingual contexts.

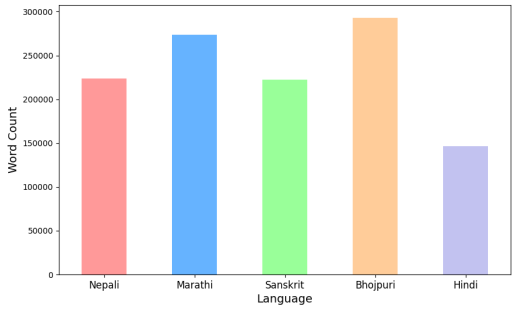
5 Experimental Results

In this section, we have discussed the results obtained while developing various models. As transformer-based models outperform ML and DL models in text classification, we have only experimented with transformer-based models. Table 3 shows the experimental results of different models on the test dataset.

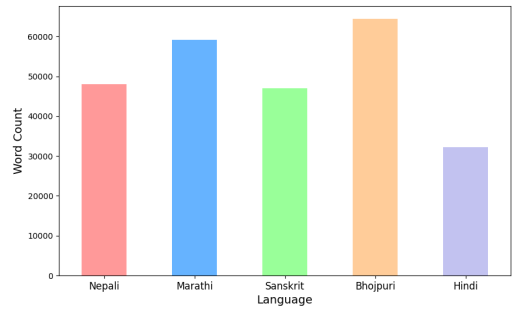
Name	Acc.	P.	R.	F1
mBERT	0.9959	0.9955	0.9953	0.9954
XLM-R	0.9959	0.9954	0.9954	0.9954
IndicBERT	0.9953	0.9947	0.9947	0.9947
Ensemble	**0.9968**	**0.9964**	**0.9966**	**0.9965**

Table 3: Results of different models on Test set

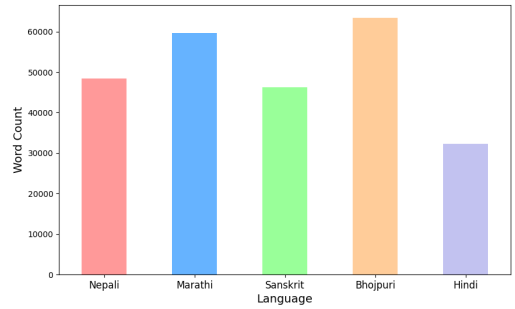
The individual models mBERT, XLM-R, and IndicBERT all have performed well with accuracies ranging from 0.9953 to 0.9959. However, the pro-



(a) Word Count Distribution in Train Set



(b) Word Count Distribution in Validation Set



(c) Word Count Distribution in Test Set

Figure 2: Word Count Distributions for Train, Validation and Test Set

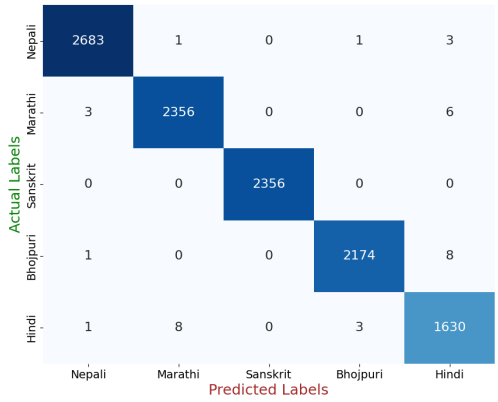


Figure 3: Confusion Matrix

posed ensemble method has outperformed them, reaching the greatest accuracy of 0.9968. The ensemble method has also been proven superior to individual models in terms of Precision, Recall, and F1 score. This is because by combining several models, biases of individual models can be reduced significantly.

6 Error Analysis

Qualitative Analysis: Individual models have faced limitations in finding language nuances. IndicBERT has struggled in low-resource cases like Bhojpuri while mBERT and XLM-R have misclassified texts due to their broader multilingual focus. These issues have affected in majority voting leading to a decrease in performance. Besides, there exist linguistic similarities among Devanagari scripted languages. Our models have been confused by these similarities and so the performance scores have been dropped. The influence of dialects and regional variations in texts have acted as a barrier against the model.

Quantitative Analysis: The confusion matrix in figure 3 analysis shows that Nepali is mostly accurately classified, with only 0.2% misclassified. Marathi has 9 misclassifications out of 2365 instances, mostly due to confusion with Hindi and Nepali. Sanskrit has no misclassifications out of 2356 instances, indicating 100% accuracy. Bhojpuri has 9 misclassifications out of 2183 instances, mostly due to confusion with Hindi and Nepali. Hindi has the highest misclassification rate, with 12 out of 1650 instances incorrectly labeled.

7 Conclusion

In this work, we have explored various transformer-based approaches for Devanagari script language identification (subtask A). We have developed an ensemble approach combining mBERT, XLM-R, and IndicBERT. Using majority voting in an ensemble approach, we have achieved an outstanding result with an F1 score of 0.9965. For our work, we have used the provided datasets in the shared task. However, after analyzing the performance, we have observed that in some cases our model has misclassified due to misclassification of individual models. In the future, we aim to try various combinations of other transformer models for the ensemble and check the performance of LLMs.

8 Limitations

The study has limitations, including the use of underrepresented dialects and informal usages of Devanagari-scripted languages in training models, and the close linguistic relationships among languages like Hindi, Marathi, Nepali, and Bhojpuri, which can lead to ambiguous cases and challenges in accurate classification. The ensemble model may struggle with sentences lacking context or code-switching. Additionally, traditional evaluation metrics like accuracy may not accurately represent the models' performance, potentially leading to an overestimated sense of effectiveness without addressing underlying weaknesses.

9 Ethical Considerations

The study's limitations include underrepresented dialects, close linguistic relationships, and potential bias. It also highlights the need for inclusivity and responsibility in future language processing endeavors, highlighting the need for data privacy and transparency.

References

Rahul Aralikatte, Miryam De Lhoneux, Anoop Kunchukuttan, and Anders Søgaard. 2021. Itihasa: A large-scale corpus for sanskrit to english translation. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 191–197.

Pramod Carpenter. 2024. 3. language identifier. *Indian Scientific Journal Of Research In Engineering And Management*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Preprint*, arXiv:1911.02116.

Hem Chandra Das and Utpal Bhattacharjee. 2024. Assamese dialect identification using static and dynamic features from vowel. *Journal of Advances in Information Technology*, 15(2).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Salman Farsi, Asrarul Eusha, Jawad Hossain, Shawly Ahsan, Avishek Das, and Mohammed Moshiul Hoque. 2024. CUET_Binary_Hackers@DravidianLangTech EACL2024: Hate and offensive language detection in Telugu code-mixed text using sentence similarity BERT. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 193–199, St. Julian's, Malta. Association for Computational Linguistics.

Md Rajib Hossain, Mohammed Moshiul Hoque, Nazmul Siddique, and M Ali Akber Dewan. 2024. Aracovtexfinder: Leveraging the transformer-based language model for arabic covid-19 text identification. *Engineering Applications of Artificial Intelligence*, 133:107987.

K Indhuja, M Indu, C Sreejith, Palakkad Sreekrishnapuram, and PR Raj. 2014. Text based language identification system for indian languages following devanagiri script. *International Journal of Engineering*, 3(4).

Farhan Ahmad Jafri, Kritesh Rauniyar, Surendrabikram Thapa, Mohammad Aman Siddiqui, Matloob Khushi, and Usman Naseem. 2024. Chunav: Analyzing hindi hate speech and targeted groups in indian election discourse. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Farhan Ahmad Jafri, Mohammad Aman Siddiqui, Surendrabikram Thapa, Kritesh Rauniyar, Usman Naseem, and Imran Razzak. 2023. Uncovering political hate speech during indian election campaign: A new low-resource dataset and baselines.

Tommi Jauhiainen, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2024. 1. introduction to language identification.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar, and Raviraj Joshi. 2021. L3cubemahasent: A marathi tweet-based sentiment analysis dataset. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 213–220.

Riya Menon. 2022. 8. detectsy: A system for detecting language from the text, images, and audio files. *International Journal For Science Technology And Engineering*.

Atul Kr Ojha. 2019. English-bhojpuri smt system: Insights from the karaka model. *arXiv preprint arXiv:1905.02239*.

Tejas Pinge, Prajwal Patil, Mayur Sherki, Aditya Nandurkar, and Prof. Ravindra Chilbule. 2023. 2. text language identification and translator. *International Journal of Advanced Research in Science, Communication and Technology*.

Karthika M R and Anu George. 2023. 5. automatic language identification from non-uniform region using bi-lstm and cnn.

Kritesh Rauniyar, Sweta Poudel, Shuvam Shiwakoti, Surendrabikram Thapa, Junaid Rashid, Jungeun Kim, Muhammad Imran, and Usman Naseem. 2023. Multi-aspect annotation and analysis of nepali tweets on anti-establishment election discourse. *IEEE Access*.

Sunita Rawat, Lakshita Werulkar, and Sagarika Jaywant. 2023. 7. text-based language identifier using multinomial naïve bayes algorithm. *International journal of next-generation computing*.

Nihar Ranja Sahoo, Gyana Prakash Beria, and Pushpak Bhattacharyya. 2024. Indicconan: A multilingual dataset for combating hate speech in indian context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22313–22321.

Khalid Saifullah, Muhammad Ibrahim Khan, Suhaima Jamal, and Iqbal H Sarker. 2024. Cyberbullying text identification based on deep learning and transformer-based language models. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, 11(1):e5–e5.

Kengatharaiyer Sarveswaran, Bal Krishna Bal, Surendrabikram Thapa, Ashwini Vaidya, and Sana Shams. 2025. A brief overview of the first workshop on challenges in processing south asian languages (chipsal). In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Arpit Sharma and BN Mithun. 2023. Deep learning character recognition of handwritten devanagari script: A complete survey. In *2023 IEEE International Conference on Contemporary Computing and Communications (InC4)*, volume 1, pages 1–6. IEEE.

Deepawali Sharma, Vivek Kumar Singh, and Vedika Gupta. 2024. Tabhate: a target-based hate speech detection dataset in hindi. *Social Network Analysis and Mining*, 14(1):190.

A. V. Sriharsha, M Jahnavi, Desai Sakethram Kousik, V. Hemanth, M G Hari, and Penchala Praveen Vasili. 2024. 6. language detection using natural language processing. *Advances in computer science research*.

Surendrabikram Thapa, Kritesh Rauniyar, Farhan Ahmad Jafri, Surabhi Adhikari, Kengatharaiyer Sarveswaran, Bal Krishna Bal, Hariram Veeramani, and Usman Naseem. 2025. Natural language understanding of devanagari script languages: Language identification, hate speech and its target detection. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Surendrabikram Thapa, Kritesh Rauniyar, Shuvam Shiwakoti, Sweta Poudel, Usman Naseem, and Mehwish Nasim. 2023. Nehate: Large-scale annotated data shedding light on hate speech in nepali local election discourse. In *ECAI 2023*, pages 2346–2353. IOS Press.

S Thara and Prabaharan Poornachandran. 2021. Transformer based language identification for malayalam-english code-mixed text. *IEEE Access*, 9:118837–118850.

Atnafu Lambebo Tonja, Mesay Gemeda Yigezu, Olga Kolesnikova, Moein Shahiki Tash, Grigori Sidorov, and Alexander Gelbuk. 2022. Transformer-based model for word level language identification in code-mixed kannada-english texts. *Preprint*, arXiv:2211.14459.

A Experimental Setup

A.1 Data Preparation

In the shared task, organizers provided a training dataset and a evaluation dataset. We merged the two datasets and split them to use as train and validation dataset. We have used 80% of the combined dataset for training and the rest for validation dataset. This merging process has created a larger dataset than the provided training dataset and thus, helped the model for better training.

A.2 Parameter Settings

The overall parameter settings used in this experiment have been described in table 4.

Parameter	Value
Epoch	5
Batch size	32
Loss Function	CrossEntropyLoss
Learning Rate	1e−3

Table 4: Parameter Configuration

A.3 Environment Setup

A personal computer with a Ryzen-9 CPU (3.00 GHz) and an NVIDIA GeForce GTX 2060 GPU has been used to run the simulation. Additionally, a Kaggle Notebook set up with a P100 GPU has been used to guarantee sufficient processing power.

B Data Preprocessing

For preprocessing, we focused on normalizing the input text by converting it to a standard Unicode format to handle variations in Devanagari script encoding. The text was then tokenized using model-specific tokenizers, such as those for mBERT, XLM-R, and IndicBERT, to break it into meaningful subword units. Additionally, padding and truncation were applied to ensure that all input sequences were of same length.