byteSizedLLM@NLU of Devanagari Script Languages 2025: Language Identification Using Customized Attention BiLSTM and XLM-RoBERTa base Embeddings

Durga Prasad Manukonda
ASRlytics
Hyderabad, India
mdp0999@gmail.com

Rohith Gowtham Kodali
ASRlytics
Hyderabad, India
rohitkodali@gmail.com

Abstract

This study explores the challenges of natural language understanding (NLU) in multilingual contexts, focusing on Devanagari-scripted languages such as Nepali, Marathi, Sanskrit, Bhojpuri, and Hindi. Language identification within these languages is complex due to their structural and lexical similarities. We present a hybrid Attention BiLSTM-XLM-RoBERTa model, achieving a state-of-the-art F1 score of 0.9974 on the test set, despite limited resources. Our model effectively distinguishes between closely related Devanagari-scripted languages, providing a solid foundation for context-aware NLU systems that enhance language-specific processing and promote inclusive digital interactions across diverse linguistic communities.

1 Introduction

In the era of rapidly expanding digital content, developing effective natural language understanding (NLU) capabilities in multilingual contexts is essential, particularly for languages using the Devanagari script, such as Nepali, Marathi, Sanskrit, Bhojpuri, and Hindi. The diversity and complexity of these languages, coupled with their shared script, present distinct challenges for language identification and moderation. Addressing this need, the Shared Task on Natural Language Understanding of Devanagari Script Languages at CHIPSAL@COLING 2025 introduces three critical subtasks to enhance the automated identification and analysis of Devanagari-scripted content in multilingual environments (Thapa et al., 2025) (Sarveswaran et al., 2025)

Subtask A: Devanagari Script Language Identification hones in on discerning the specific language within Devanagari-scripted text. In multilingual digital spaces, accurately identifying the language is a prerequisite for effective processing, enabling robust multilingual NLU systems. Given a sentence in Devanagari script, this subtask's objective is to determine whether the language is Nepali, Marathi, Sanskrit, Bhojpuri, or Hindi, meeting the pressing need for accurate language differentiation among closely related languages that share the Devanagari script. This foundational task supports precise language identification, empowering deeper analysis and tailored content moderation across Devanagari-scripted languages.

Our hybrid architecture, combining an attention-based BiLSTM with XLM-RoBERTa embeddings, effectively captures the syntactic and semantic nuances required for accurate language differentiation. The BiLSTM component, enhanced by attention, improves sequential modeling, while XLM-RoBERTa provides robust multilingual embeddings. This integration enables high precision in language identification and lays a foundation for more advanced multilingual NLU. Additionally, the model's attention mechanism allows it to focus on language-specific features, further enhancing its ability to distinguish closely related Devanagari-scripted languages.

2 Related Work

Two studies focus on language identification for Indian languages in the Devanagari script. The first uses n-gram models to classify languages like Hindi, Marathi, and Sanskrit based on character- and word-level frequency patterns Indhuja. et al. (2014). The second applies machine learning and deep learning to capture subtle lexical differences in poetry Acharya et al. (2020). Both highlight progress in language identification and the challenges of linguistic similarities and stylistic variations.

Expanding to native and romanized forms, proposed Madhani et al. (2023), using FastText and IndicBERT to identify 22 Indic languages. Together, these studies illustrate advancements and ongoing challenges in distinguishing related languages and

Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), pages 248–252
January 19, 2025.

text styles in Devanagari.

3 Dataset & Task

Task A focuses on identifying the specific Devanagari-scripted language of a given text, with a dataset comprising samples from five 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., 2024)(Jafri et al., 2023). Accurate language classification in multilingual contexts relies heavily on this task. To facilitate training and evaluation, the dataset is divided into training, validation, and test sets.

The dataset consists of sentences in five Devanagari-script languages, with labels assigned as follows: 'Nepali' is labeled as '0', 'Marathi' as '1', 'Sanskrit' as '2', 'Bhojpuri' as '3', and 'Hindi' as '4', allowing for efficient and accurate language classification.

Table 1 provides a detailed analysis of language distribution within the training and validation sets, highlighting representation across subsets. The curated and labeled data supports NLP tasks in Devanagari-script languages, forming a foundation for robust language differentiation.

Class	Train	Valid	Test
Nepali (0)	12543	2688	2688
Marathi (1)	11034	2364	2365
Sanskrit (2)	10996	2356	2356
Bhojpuri (3)	10184	2182	2183
Hindi (4)	7659	1642	1642
Total	52416	11232	11234

Table 1: Distribution of samples in Train, Validation(Valid) and Test datasets for each class in SubTask A

Additionally, we curated datasets to fine-tune multilingual RoBERTa (xlm-roberta-base) on masked language modeling for five languages, following the approach of Joshi (2022): Bhojpuri (9.3MB from GitHub[1]), Nepali (50MB from Kaggle[2]), Sanskrit (50MB from Kaggle[3]), and Hindi and Marathi (50MB each from AI4Bharat[4]).

[1] https://github.com/shashwatup9k/bho-resources

[2] https://www.kaggle.com/datasets/lotusacharya/nepalinewsdataset

[3] https://www.kaggle.com/datasets/rushikeshdarge/sanskrit

[4] https://github.com/AI4Bharat/indicnlp_corpus

4 Methodology

This study presents a hybrid Attention BiLSTM-XLM-RoBERTa model, inspired by Hochreiter and Schmidhuber (1997); Conneau et al. (2019); Manukonda and Kodali (2024a); Kodali and Manukonda (2024); Manukonda and Kodali (2024b); Brauwers and Frasincar (2023), for language identification within the Devanagari script. As shown in Figure 1, the model integrates deep contextual embeddings from the fine-tuned masked language model (MLM) of XLM-RoBERTa with a bidirectional LSTM and attention mechanism to enhance language-specific feature extraction. Each model component and its mathematical foundation are detailed below.

The input sequence is first passed to XLM-RoBERTa base, generating contextualized embeddings $\mathbf{X} \in R^{T \times D}$, where $D = 768$ represents the embedding dimension:

$$\mathbf{X} = \mathbf{XLMRoBERTa}(input\_ids, attention\_mask) \quad (1)$$

These embeddings are fed into a BiLSTM to capture sequential dependencies, producing bidirectional hidden states $\mathbf{H}fwd$ and $\mathbf{H}bwd$, which combine as:

$$\mathbf{H}t = [\mathbf{H}fwd, t; \mathbf{H}_{bwd,t}] \quad (2)$$

An attention mechanism then assigns relevance to each $\mathbf{H}_t$, yielding attention weights α_t:

$$\mathbf{a}t = \tanh(\mathbf{W}att \cdot \mathbf{H}_t), \quad \alpha_t = \frac{\exp(\mathbf{a}t)}{\sum t = 1^T \exp(\mathbf{a}_t)} \quad (3)$$

The attention-weighted representation $\mathbf{H}_{attended}$ is computed as:

$$\mathbf{H}attended = \sum t = 1^T \alpha_t \cdot \mathbf{H}_t \quad (4)$$

Layer normalization and dropout are optional residual components that help mitigate overfitting and stabilize training, especially in complex language scenarios. They enhance generalization by reducing variance and stabilizing weight updates, benefiting smaller or noisier datasets. To combat overfitting, $\mathbf{H}_{attended}$ undergoes layer normalization and dropout:

$$\mathbf{H}_{dropout} = Dropout(LayerNorm(\mathbf{H}_{attended})) \quad (5)$$

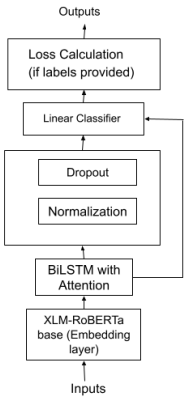

Figure 1: Architecture of the BiLSTM-XLM-RoBERTa Classifier Model. Layer normalization and dropout regularization enhance generalization, especially for smaller or noisier datasets.

Finally, $\mathbf{H}_{dropout}$ is passed through a classification layer to produce logits:

$$\textbf{logits} = \mathbf{W}_{cls} \cdot \mathbf{H}_{dropout} + \mathbf{b}_{cls} \tag{6}$$

During training, cross-entropy loss L is calculated between predicted logits and true labels:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{7}$$

This hybrid model leverages XLM-RoBERTa base embeddings, BiLSTM sequential processing, and attention for precise language differentiation in Devanagari-scripted multilingual contexts.

5 Experiment Setup

Our experiment setup involved data preprocessing, model fine-tuning, and architecture optimization to assess language identification across Devanagari-scripted languages, evaluated by accuracy and Macro F1 scores on the validation dataset.

Our unique setup involved tokenizing and normalizing datasets for compatibility with the XLM-RoBERTa base model, adapting samples to the Devanagari script. Fine-tuning on masked language modeling (MLM) used a 15% masking ratio and a learning rate of 2×10^{-5}, achieving a perplexity score of 5.33 over 7 epochs, indicating effective contextual adaptation.

Following extensive classifier testing, we selected the Attention BiLSTM-XLM-RoBERTa architecture(Figure 1) for its superior performance. This model incorporates a BiLSTM layer (hidden size 256, 2 LSTM layers, dropout 0.3) to capture sequential dependencies and an attention mechanism to emphasize language-specific features. The setup was fine-tuned over 6 epochs with a learning rate of 1×10^{-5}, using optional residual layers for normalization and dropout to enhance stability and mitigate overfitting.

This setup provides a comprehensive framework to evaluate the impact of model fine-tuning, architecture, and data preparation on multilingual classification within the Devanagari script.

6 Results and Discussion

Table 2 summarizes the performance of various classifiers using fine-tuned XLM-RoBERTa base embeddings for language identification within Devanagari-scripted languages. Initially, we experimented with several traditional linear classifiers; however, our hybrid Attention BiLSTM-XLM-RoBERTa model achieved the best performance on the validation set, leading us to proceed with this architecture for the test set.

The **Attention BiLSTM-XLM-RoBERTa** model outperformed all classifiers, achieving an accuracy of 0.9986 and a Macro F1-score of 0.9984 on the validation set, and 0.9976 accuracy with a 0.9974 Macro F1-score on the test set. This superior performance highlights the effectiveness of combining XLM-RoBERTa's contextual embeddings with BiLSTM and attention mechanisms, enabling a nuanced focus on language-specific features. The high scores indicate robust language identification, capturing syntactic and semantic nuances, even among closely related languages. While other classifiers using fine-tuned XLM-RoBERTa embeddings performed well, the hybrid model provided a clear advantage.

Table 3 provides a comparison of the top 5 ranked scores on the SubTask A test set, where our team, **byteSizedLLM**, achieved 5th place with an F1-score of 0.9974. This ranking reaffirms the model's effectiveness and positions it competitively within the overall landscape.

Overall, our findings show that the hybrid Attention BiLSTM-XLM-RoBERTa architecture, combining BiLSTM with transformer-based embeddings, provides a significant advantage. Fine-tuning XLM-RoBERTa's MLM on task-specific data further improved performance. This approach underscores the value of integrating bidirectional embeddings and attention mechanisms for precise

Classifier	Val Acc	Val F1	Test Acc	Test F1
XLM-RoBERTa base (Transformers)	0.9972	0.9969	0.9970	0.9964
XGBoost (xgb)	0.9962	0.9957	0.9945	0.9939
Random Forest (rf)	0.9962	0.9957	0.9942	0.9936
Logistic Regression (lr)	0.9971	0.9967	0.9961	0.9957
Gradient Boosting (gb)	0.9954	0.9948	0.9933	0.9926
Support Vector Classifier (svc)	0.9969	0.9965	0.9954	0.9949
AdaBoost (ada)	0.9562	0.9514	0.9564	0.9520
Extra Trees (extra_trees)	0.9955	0.9950	0.9943	0.9937
Ridge Classifier (ridge)	0.9950	0.9944	0.9944	0.9937
Stochastic Gradient Descent (sgd)	0.9974	0.9971	0.9955	0.9950
Ensemble (xgb,lr, rf, svc, sgd)	0.9970	0.9967	0.9955	0.9949
Attention BiLSTM-XLM-RoBERTa	**0.9986**	**0.9984**	**0.9974**	**0.9976**

Table 2: Comparison of Validation (Val) and Test Accuracies (Acc) and Macro-F1 Scores (F1) Across Different Classifiers

Team Name	F1-Score	Rank
CUFE	0.9997	1
CLTL	0.9982	2
1-800-SHARED-TASKS	0.9979	3
1-800-SHARED-TASKS	0.9976	4
byteSizedLLM	0.9974	5

Table 3: Comparison of Top 5 Ranked Scores on the SubTask A Test Set

language differentiation in multilingual Devanagari contexts. A key limitation was the scarcity of open-source datasets for fine-tuning MLM and computational constraints limiting us to the base model. Future exploration with larger models could further improve language identification

7 Conclusion and Future work

This study presented an innovative hybrid approach, combining the XLM-RoBERTa base embeddings with traditional classifiers and an Attention BiLSTM architecture for effective language identification in Devanagari-scripted multilingual contexts. Our proposed Attention BiLSTM-XLM-RoBERTa model achieved top performance among all classifiers tested, yielding high accuracy and Macro F1 scores, and ultimately ranking 5th overall with minimal differences from the top entries. These findings underscore the strength of integrating transformer-based embeddings with sequential and attention mechanisms, highlighting the potential of this approach to capture language-specific nuances even with limited MLM fine-tuning data and computational resources.

Further fine-tuning MLM on a larger dataset and scaling to XLM-RoBERTa-large could improve embedding quality and capture nuanced language variations. This research underscores the importance of robust embeddings with attention mechanisms for language-specific features, advancing Devanagari multilingual NLP capabilities.

8 Limitations and Ethical Considerations

8.1 Limitations

The Attention BiLSTM-XLM-RoBERTa model demonstrated strong performance but has limitations affecting generalizability. Using XLM-RoBERTa base may restrict the model's ability to capture complex contextual nuances across diverse languages. Computational constraints prevented exploring larger XLM-RoBERTa variants, potentially limiting performance gains, and limited data for fine-tuning the masked language model (MLM) may affect robustness, particularly for underrepresented languages in the Devanagari script family.

8.2 Ethical Considerations

This study prioritizes inclusivity for low-resource Devanagari-scripted languages, recognizing the potential impacts on linguistic communities. To address concerns of bias and fairness, we conduct regular evaluations of training data and model outputs, promote responsible interpretation and implementation of model outputs, and carefully consider community impact. These measures aim to support developing fair and inclusive language technologies.

References

Priyankit Acharya, Aditya Ku. Pathak, Rakesh Ch. Balabantaray, and Anil Ku. Singh. 2020. Language identification of devanagari poems. *Preprint*, arXiv:2012.15023.

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.

Gianni Brauwers and Flavius Frasincar. 2023. A general survey on attention mechanisms in deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3279–3298.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

K Indhuja., M. G. Indu, C. Sreejith, and Purushottam Raj. 2014. Text based language identification system for indian languages following devanagiri script. *International journal of engineering research and technology*, 3.

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.

Raviraj Joshi. 2022. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Rohith Kodali and Durga Manukonda. 2024. byteSizedLLM@DravidianLangTech 2024: Fake news detection in Dravidian languages - unleashing the power of custom subword tokenization with Subword2Vec and BiLSTM. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 79–84, St. Julian's, Malta. Association for Computational Linguistics.

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.

Yash Madhani, Mitesh M. Khapra, and Anoop Kunchukuttan. 2023. Bhasha-abhijnaanam: Native-script and romanized language identification for 22 indic languages. *Preprint*, arXiv:2305.15814.

Durga Manukonda and Rohith Kodali. 2024a. byteLLM@LT-EDI-2024: Homophobia/transphobia detection in social media comments - custom subword tokenization with Subword2Vec and BiLSTM. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 157–163, St. Julian's, Malta. Association for Computational Linguistics.

Durga Prasad Manukonda and Rohith Gowtham Kodali. 2024b. Enhancing multilingual natural language processing with custom subword tokenization: Subword2vec and bilstm integration for lightweight and streamlined approaches. In *2024 6th International Conference on Natural Language Processing (ICNLP)*, pages 366–371.

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

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*.

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)*.

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.