CUET_Big_O@NLU of Devanagari Script Languages 2025: Identifying Script Language and Detecting Hate Speech Using Deep Learning and Transformer Model

Md. Refaj Hossan*, **Nazmus Sakib***, **Md. Alam Miah**
Jawad Hossain and Mohammed Moshiul Hoque
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
{u1904007,u1904086,u1904102,u1704039}@student.cuet.ac.bd
moshiul_240@cuet.ac.bd

Abstract

Text-based hate speech has been prevalent and is usually used to incite hostility and violence. Detecting this content becomes imperative, yet the task is challenging, particularly for low-resource languages in the Devanagari script, which must have the extensive labeled datasets required for effective machine learning. To address this, a shared task has been organized for identifying hate speech targets in Devanagari-script text. The task involves classifying targets such as individuals, organizations, and communities and identifying different languages within the script. We have explored several machine learning methods such as LR, SVM, MNB, and Random Forest, deep learning models using CNN, BiLSTM, GRU, CNN+BiLSTM, and transformer-based models like Indic-BERT, m-BERT, Verta-BERT, XLM-R, and MuRIL. The CNN with BiLSTM yielded the best performance (F1-score of 0.9941), placing the team 13th in the competition for script identification. Furthermore, the fine-tuned MuRIL-BERT model resulted in an F1 score of 0.6832, ranking us 4th for detecting hate speech targets.

1 Introduction

Digital platforms such as Facebook, Instagram, and YouTube have emerged as a common medium for public expression with the rapid expansion of online communication. Unfortunately, these digital platforms also act as conduits for injurious content, including hate speech, which fosters hostility and marginalization of communities and threatens social cohesion. Hateful content can attack social harmony based on race, gender, religion, nationality, political support, immigration status, and personal beliefs (Paz et al., 2020). Hence, determining whether shared content on social media is hateful is crucial.

While much recent work has focused on identifying hate and offensive content in high-resource languages such as English, Spanish (del Arco et al., 2021), and Arabic (Omar et al., 2020), which have abundant linguistic resources and datasets available, the challenge remains in low-resource settings where effective hate speech detection is obstructed due to a lack of resources (Magueresse et al., 2020). Hence, it is also crucial in a multilingually rich region like South Asia, where multiple languages and scripts are used daily. In this context, the identification of hate speech is essential in the Devanagari script, which encompasses languages such as Hindi, Marathi, Nepali, and Sanskrit, each with millions of speakers. Moreover, the complex structure of Devanagari, with frequent code-mixing and nuanced expressions, makes it challenging to distinguish between languages. At the same time, detecting hate speech requires culturally adept models able to estimate indirect or inexact language. Concentrated on the circumstances, the organizers (Thapa et al., 2025) presented different datasets for three subtasks by combining several datasets (Jafri et al., 2023, 2024; Thapa et al., 2023; Rauniyar et al., 2023; Ojha, 2019; Kulkarni et al., 2021; Aralikatte et al., 2021) for identifying Devanagari script language in subtask A, hate speech detection in subtask B, and target identification for hate speech in subtask C in the first workshop on Challenges in Processing South Asian Languages (CHiPSAL) (Sarveswaran et al., 2025). However, this work aims to outline the contributions to subtasks A and C, which are as follows:

- Developed a hybrid model using CNN with BiLSTM for Devanagari script identification and fine-tuned MuRIL for target hate speech detection in the Devanagari script language.
- Explored various Machine Learning (ML), Deep Learning (DL), and transformer-based

*Authors contributed equally to this work.

Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), pages 253–259
January 19, 2025.

models to identify Devanagari script language and target identification for hate speech.

- Investigated and contrasted multiple performance metrics and in-depth error analysis for the models to perceive the best strategy toward identifying Devanagari script language and classifying target hate speech.

2 Related Work

Earlier efforts involved traditional ML algorithms to segregate the script and identify the language; these laid a platform for more advanced techniques in this area. For instance, KumarShrivastava and Chaurasia (2012) obtained a 100% recognition rate of Devanagari characters using SVM with polynomial kernel by testing different kernels and segment count on their dataset. A survey conducted by Jayadevan et al. (2011) reviewed the state-of-the-art techniques concerning machine-printed and handwritten Devanagari OCR by underlining different feature extraction methods and classification models. Moreover, Halder et al. (2015) presented their analysis of Devanagari characters for writer identification with several techniques and achieved 99.12% accuracy with LIBLINEAR. Another work focused on script identification from Indian documents such as Bangla, Devanagari, Gujarati, etc., using feature extraction techniques like Log-Gabor Filtering and achieved 97.11% accuracy with ten different Indian scripts using optimized KNN technique (Joshi et al., 2006).

While Devanagari script identification is going on, hate speech detection is also of prime importance in research because it segregates social unity, and lots of research is being conducted for high-resource languages. Fortuna and Nunes (2018) gave a comprehensive overview of the hate speech detection techniques, pinpointing the need for approaches tailored for multilingual contexts. Therein, Nandi et al. (2024) presented a review of recent research on hate speech detection in Indian languages, discussed the challenges, and then analyzed various methodologies, datasets, and results to show the gaps and opportunities for future work in this critical area of study. Another work done by Jha et al. (2020) proposed a FastText-based model for the Hindi Language to classify offensive and non-offensive texts, and an accuracy of 92.2% has been achieved by grid-search hyperparameter tuning using the Devanagari Hindi Offensive Tweets (DHOT) dataset[1]. The existing research contributions in hate speech detection, addressing types related to racism, sexism, and religious hate, and the methods developed for mitigating them, along with the identification of challenges, have been reviewed by Parihar et al. (2021). Furthermore, several works have been performed to detect hate speech in code-mixed Hindi-English (Chopra et al., 2023), code-switched Hindi-English (Sharma et al., 2022) by using deep learning, transformer-based approaches to obtain superior performance. Prior work has yet to focus on target-specific hate speech detection (individual, community, organization) in Devanagari script with code-mixed language. In this context, our work introduces not only the model for it but also proposes a script identification component specific to Devanagari by addressing the complexity of the script and challenges posed due to code-mixing in South Asian languages.

3 Task and Dataset Description

In the shared task (Thapa et al., 2025), there were three subtasks: A, B, and C. However, the goal of subtask A was to identify whether the language given in the dataset belongs to Nepali, Marathi, Sanskrit, Bhojpuri, or Hindi, making it a multi-class classification problem. Along with subtask A, the objective of subtask C was to identify the target of the hate speech, categorized as either *Individual*, *Organization*, or *Community* classes in Devanagari script language. For subtask A, the train, valid, and test datasets comprise 52422, 11233, and 11234 texts, respectively. Class-wise samples and dataset statistics are provided in Table 1.

Table 1: Class-wise distribution of train, validation, and test set for subtask A, where W_T and UW_T denote total words in three datasets and total unique words in train set respectively

Classes	Train	Valid	Test	W_T	UW_T
Nepali	12544	2688	2688	320384	26536
Marathi	11034	2364	2365	392735	32332
Sanskrit	10996	2356	2356	315875	64652
Bhojpuri	10184	2182	2183	420856	15779
Hindi	7664	1643	1642	211029	8933
Total	52422	11233	11234	1660879	148232

For subtask C, the train, validation, and test datasets consist of 2214, 474, and 475 texts, and the datasets are imbalanced. Table 2 provides the

[1] https://github.com/vikaskumarjha9/hindi_abusive_dataset

class-wise samples and dataset statistics. The implementation details of the tasks will be found in the GitHub repository[2].

Table 2: Class-wise distribution of train, validation, and test set for subtask C, where W_T and UW_T denote total words in three datasets and total unique words in train data respectively

Classes	Train	Valid	Test	W_T	UW_T
Individual	1074	230	230	42438	11963
Organization	856	183	184	11586	8891
Community	284	61	61	10564	3931
Total	2214	474	475	64588	24785

4 Methodology

Several ML, DL, and transformer-based models were explored to develop the baselines as depicted in Figure 1.

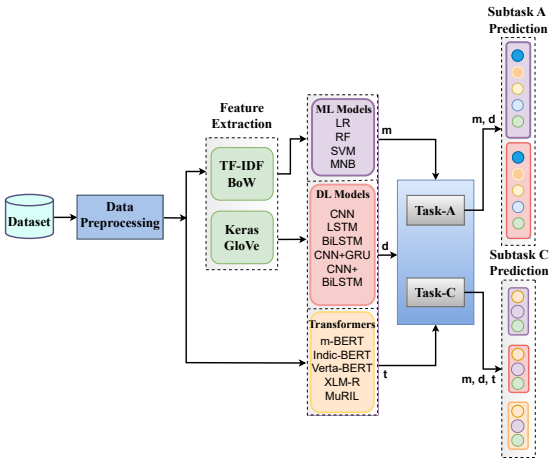

Figure 1: Schematic process of Devanagari script identification and target identification for hate speech

4.1 Data Preprocessing

Since the dataset originates in Devanagari, aggregated from several sources, by default, it contains quite a large amount of irrelevant and duplicate data. Therefore, the first significant work was extensive preprocessing of data. It included the removal of emojis, symbols, signs, numbers, and extra punctuation marks from the text. Data augmentation techniques have not been used, as more emphasis was put on cleaning and refining the data to prepare them for appropriate model training.

[2] https://github.com/RJ-Hossan/CHIPSAL-25

4.2 Feature Extraction

Feature extraction in NLP transforms raw text into numerical values for machine learning and deep learning models. Our approach extracts unigram and bigram features using TF-IDF for machine learning algorithms. For deep learning, text preprocessing involves tokenization via the Tokenizer class of TensorFlow Keras[3], which handles out-of-vocabulary words using placeholder tokens. These tokens are passed into an Embedding layer, converting them into dense vector representations. Additionally, we incorporate pre-trained GloVe embeddings, which map each word to a 100D vector, with an embedding matrix of shape (10,000, 100) for the top 10,000 words in the tokenizer's vocabulary.

4.3 Machine Learning Models

We have employed various machine learning (ML) models to identify instances of hate speech. Specifically, we employed Logistic Regression (LR), Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), Random Forest (RF), Gradient Boosting (GB). In subtask C, we have also employed hyperparameter tuning e.g., linear and RBF kernel for SVM, different learning rate for LR, GB, different estimator values, LIBLINEAR (Fan et al., 2008) solver function for LR, etc., using GridSearchCV[4] to get the superior performance.

4.4 Deep Learning Models

For deep learning models, we considered several approaches, such as LSTM, BiLSTM, CNN, CNN + BiLSTM, and CNN + GRU. These models were trained with tokenization and embedding techniques. The hybrid BiLSTM with CNN model is configured with a maximum vocabulary size of 10,000, a sequence length of 100, and an embedding dimension of 128. It has a CNN branch with 64 filters of size 5 and a BiLSTM branch with 64 units, trained over 45 epochs with a batch size of 32. Using sparse categorical cross-entropy as the loss function and the 'Adam' optimizer at a learning rate of 1e-4, the class imbalance was addressed by computing class weights. At the same time, the training was optimized through Reduce Learning Rate and Early Stopping callbacks for improved performance in Subtask A. Table 3 shows the fine-

[3] https://www.tensorflow.org/api_docs/python/tf/keras/layers/Embedding

[4] https://scikit-learn.org/dev/modules/generated/sklearn.model_selection.GridSearchCV.html

tuned hyperparameters for the deep learning-based models for subtask A.

Table 3: Model configuration for subtask A

Parameter	Value
Vocabulary Size	10,000
Sequence Length	100
Embedding Dimension	128
CNN Filters	64 filters of size 5
BiLSTM Units	64
Epochs	45
Batch Size	32
Optimizer	Adam
Learning Rate	1e-4

4.5 Transformer-Based Models

With the mechanism of attention embedding, transformer-based models efficiently process large-scale contextual information and, therefore, prove ideal for multilingual and cross-lingual tasks. To accomplish the tasks, we explored several transformer-based models such as m-BERT (Pires et al., 2019), Indic-BERT (Dabre et al., 2022), MuRIL-BERT (Khanuja et al., 2021), and XLM-R (Conneau et al., 2020) to study their performances for a diverse range of linguistic settings. Each of these models had been fine-tuned to the respective classification tasks. In the MuRIL-BERT model, hyperparameter tuning was done by fixing the batch size to 8, the learning rate to 2e-4, and modifying the weight decay to 0.06. Then, after training for 13 epochs, optimal performance was reached. Due to more robust regularization from an increased weight decay of 0.01 to 0.06, the training loss reduced when the model's generalization improved. Table 4 shows the fine-tuned hyperparameters for the transformer-based models for subtask C.

Table 4: Model configuration for subtask C

Parameter	Value
Batch Size	8
Epochs	13
Weight Decay	0.06
Learning Rate	2e-4

4.6 Computational Requirements

The model was trained on a dual GPU setup (NVIDIA Tesla T4x2), using parallel processing for BiLSTM, convolution, and transformer layers. The BiLSTM+CNN model used 5-8 GB of GPU memory, while MuRIL-BERT required 20 GB. Training for 45 epochs of BiLSTM+CNN and 13 epochs of MuRIL-BERT took 45-60 minutes, depending on dataset size and class weight calculations, balancing computational efficiency and performance.

5 Result Analysis

Table 5 compares classifier performance in two subtasks, showing differences in precision, recall, and F1-score. Traditional models for Devanagari script identification, such as LR and SVM, produce high F1-scores of 0.9628 and 0.9531, respectively, but fall somewhat short of deep learning models. Among these, the lowest performing remains RF, with an F1-score of 0.9368. Finally, LSTM and BiLSTM outperformed the neural networks by the classical approaches, achieving F1-scores of 0.9791 and 0.9917, respectively. The CNN and CNN + GRU models achieved F1-scores of 0.9916 and 0.9915, respectively, while CNN + BiLSTM outperformed them with a near-perfect F1-score of 0.9941 (subtask A).

Table 5: Result comparison on test data, where P, R, and F1 denote precision, recall, and F1-score, respectively, and K and G represent Keras and GloVe embeddings

Classifiers	Script Identification			Target Hate Speech		
	P	R	F1	P	R	F1
LR	0.9628	0.9628	0.9628	0.61	0.53	0.54
SVM	0.9540	0.9524	0.9531	0.59	0.48	0.46
RF	0.9382	0.9359	0.9368	0.76	0.49	0.46
MNB	0.9511	0.9424	0.9454	0.56	0.45	0.43
CNN (K)	0.9916	0.9917	0.9916	0.57	0.55	0.56
LSTM (K)	0.9789	0.9797	0.9791	0.50	0.48	0.48
BiLSTM (K)	0.9917	0.9917	0.9917	0.48	0.47	0.47
CNN + GRU (K)	0.9917	0.9913	0.9915	0.49	0.48	0.48
CNN + BiLSTM (G)	0.7024	0.5789	0.5146	0.49	0.40	0.37
CNN + BiLSTM (K)	**0.9941**	**0.9940**	**0.9941**	0.48	0.46	0.47
Indic-BERT	-	-	-	0.61	0.61	0.61
m-BERT	-	-	-	0.61	0.60	0.60
verta-BERT	-	-	-	0.61	0.60	0.61
XLM-R	-	-	-	0.65	0.71	0.66
MuRIL-BERT	-	-	-	**0.68**	**0.68**	**0.68**

For the target hate speech detection subtask, precision, recall, and F1-scores dropped across models, reflecting the task's complexity. Traditional models performed far worse, with the best among them, LR, achieving an F1-score of only 0.54, while MNB had the lowest performance with an F1-score of just 0.43. The DL-based models also showed relatively poor F1 scores, ranging from 0.37 to 0.56. Transformer-based models performed much better on this task; specifically, MuRIL-BERT had the highest F1-score of 0.68, outperforming XLM-R (0.66) and m-BERT (0.60), helped us to rank 4th in subtask C. Interestingly, both Indic-BERT and verta-BERT achieved F1-scores of 0.61,

reinforcing the trend that transformer-based models consistently outperformed traditional and neural network-based classifiers in the nuanced task of hate speech detection.

Appedix A provides a comprehensive error analysis of the proposed models, examining their performance in identifying the Devanagari script and detecting hate speech targets.

6 Conclusion

This paper introduced techniques of Devanagari script identification and target hate speech detection. This research bridges technology with linguistic diversity, creating a more inclusive digital world. The results demonstrated that the hybrid CNN with BiLSTM model outperformed other ML and DL models for script identification tasks by achieving the highest F1-score of 0.9941. At the same time, MuRIL-BERT performed best among all other models in target hate speech detection with an F1-score of 0.68. However, the integration of transformer-based models might perform even better for script identification. Therefore, in the future, we will explore other word embedding techniques and contextualized embeddings like GPT and ELMo in these tasks for enhancing performance for Devanagari script identification and target hate speech detection. Furthermore, ensemble methods combining several transformers with various fusion models designed for specific tasks can improve the results.

7 Limitations

The current work on script identification and target hate speech detection has several drawbacks, influenced by the following factors:

- Pre-trained transformer models may fail when the context differs significantly from their training data.
- The resort to DL models employed did not give the anticipated result. This indicates that other embeddings should be tried, and better models must be devised.
- Overall, this work is limited by dataset imbalance, reliance on existing models without architectural innovation, moderate hate speech detection performance, particularly in capturing subtle contextual cues, and a lack of advanced data augmentation techniques to address class imbalance.

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.

Abhishek Chopra, Deepak Kumar Sharma, Aashna Jha, and Uttam Ghosh. 2023. A framework for online hate speech detection on code-mixed hindi-english text and hindi text in devanagari. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(5):1–21.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh Khapra, and Pratyush Kumar. 2022. Indicbart: A pre-trained model for indic natural language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics.

Flor Miriam Plaza del Arco, María Dolores Molina-González, L. Alfonso Ureña-López, and María Teresa Martín-Valdivia. 2021. Comparing pre-trained language models for spanish hate speech detection. *Expert Syst. Appl.*, 166:114120.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. 9:1871–1874.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Chayan Halder, Kishore Thakur, Santanu Phadikar, and Kaushik Roy. 2015. Writer identification from handwritten devanagari script. In *Information Systems Design and Intelligent Applications: Proceedings of Second International Conference INDIA 2015, Volume 2*, pages 497–505. Springer.

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.

R Jayadevan, Satish R Kolhe, Pradeep M Patil, and Umapada Pal. 2011. Offline recognition of devanagari script: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):782–796.

Vikas Kumar Jha, Hrudya P, Vinu P N, Vishnu Vijayan, and Prabaharan P. 2020. Dhot-repository and classification of offensive tweets in the hindi language. *Procedia Computer Science*, 171:2324–2333. Third International Conference on Computing and Network Communications (CoCoNet'19).

Gopal Datt Joshi, Saurabh Garg, and Jayanthi Sivaswamy. 2006. Script identification from indian documents. In *Document Analysis Systems VII: 7th International Workshop, DAS 2006, Nelson, New Zealand, February 13-15, 2006. Proceedings 7*, pages 255–267. Springer.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vishnu Subramanian, and Partha Pratim Talukdar. 2021. Muril: Multilingual representations for indian languages. *ArXiv*, abs/2103.10730.

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.

Shailendra KumarShrivastava and Pratibha Chaurasia. 2012. Handwritten devanagari lipi using support vector machine. *International Journal of Computer Applications*, 43:20–25.

Alexandre Magueresse, Vincent Carles, and Evan Heetderks. 2020. Low-resource languages: A review of past work and future challenges. *ArXiv*, abs/2006.07264.

Arpan Nandi, Kamal Sarkar, Arjun Mallick, and Arkadeep De. 2024. A survey of hate speech detection in indian languages. *Social Network Analysis and Mining*, 14(1):70.

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

Ahmed Omar, Tarek M. Mahmoud, and Tarek Abd-El-Hafeez. 2020. Comparative performance of machine learning and deep learning algorithms for arabic hate speech detection in osns. In *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)*, pages 247–257, Cham. Springer International Publishing.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

María Antonia Paz, Julio Montero-Díaz, and Alicia Moreno-Delgado. 2020. Hate speech: A systematized review. *Sage Open*, 10(4):2158244020973022.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

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)*.

Arushi Sharma, Anubha Kabra, and Minni Jain. 2022. Ceasing hate with moh: Hate speech detection in hindi–english code-switched language. *Information Processing & Management*, 59(1):102760.

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.

A Appendix

We have performed both quantitative and qualitative error analysis in order to obtain in-depth insights into the performance of the proposed model.

Quantitative Analysis: The best performing models were used to conduct a quantitative error analysis, utilizing confusion matrices shown in Figure A.1 and Figure A.2 for subtasks A and C, respectively.

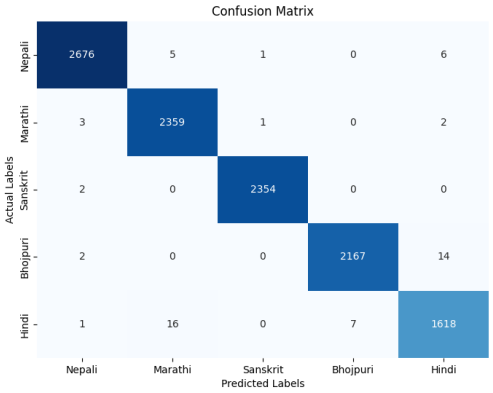

Figure A.1: Confusion matrix of the proposed model (CNN+BiLSTM) for subtask A

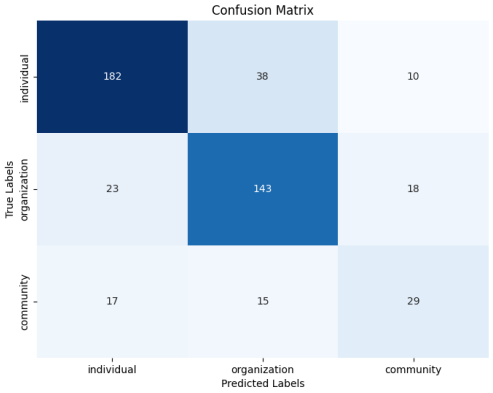

Figure A.2: Confusion matrix of the proposed model (finetuned MuRIL) for subtask C

The proposed hybrid CNN with BiLSTM model for subtask A perfectly classified 11,174 instances among 11,234 samples with very negligible misclassifications in Nepali and Sanskrit. It performed badly in distinguishing similar languages by mislabeling 16 Hindi samples as Marathi and 7 as Bhojpuri. Additionally, the fine-tuned MuRIL model performed well on target hate speech detection, wherein it rightly classified 182 out of 230 instances of *Individual* and 143 out of 184 instances of *Organization*. However, it misclassified 23 *Organization* and 38 *Individual* instances. The more difficult class was *Community* with only 61 instances; only 29 were classified correctly, mostly confused with *Individual* or *Organization*. This may happen due to the difficulty in distinguishing targets, arising from linguistic overlap in community-targeted speech and the subtlety of contextual cues.

Qualitative Analysis: Figure A.3 portrays predicted outputs for sample inputs of the proposed model for the Devanagari script identification task. It correctly predicted the text samples 1, 2, and 3, but incorrectly predicted text sample 4 as Hindi instead of Nepali. Figure A.4 represents the qualitative analysis of the proposed MuRIL-BERT model on target hate speech detection. Our proposed model correctly predicted text samples 1 and 3 but wrongly predicted samples 2 and 4. These are probably related to class imbalance, considering the *Community* class has fewer instances, 284, compared to the rest of the classes.

Sample Text	Actual Label	Predicted Label
Sample 1: निर्वाचन परिणाम ले बल्ल बुद्धि आयो?	Nepali	Nepali
Sample 2: ततो गजगतो राजा भगदत्तः प्रतापवान्। अर्जुनं शरवर्षेण वारयामास संयुगे॥ अर्जुनस्तु ततो नागमायान्तं रजतोपमैः। विमलैरायसैस्तीक्ष्णै रविध्यत महारणे॥	Sanskrit	Sanskrit
Sample 3: अबहीं चुनाव में हार के दरद कमो ना भइल रहे कि मोदी के राजनीति के तिरशूल आ के करेजा में धँसि गइल	Bhojpuri	Bhojpuri
Sample 4: कांग्रेस भर्ती व्यापार।	Nepali	Hindi

Figure A.3: Few examples of predicted outputs by the proposed method (CNN + BiLSTM) for subtask A

Sample Text	Actual Label	Predicted Label
Sample 1: कस्ता पार्टी सभापति हुन्, हँसियाहथौडामा भोट नहाले कारबाही रे ।\n# मारेके	Individual	Individual
Sample 2: अहिलेको चुनाव को परिणाम हेर्दा ,\nनेपाल विकास न हुनुमा जति भ्रष्ट नेता हरु र भ्रष्ट कर्मचारीहरु हाथ छ,\n त्यो भन्दा बढी हाथ नेपाली जनताहरुको देखिन्छ।\n 😭😭😭😭	Individual	Community
Sample 3: @nirab_argument @Gk1402Chitwan @Chitwoney यस्तो बेला पनि भरतपुरमा एमालेले जीतेन भने, एमाले पार्टी खारौज गर्दै हुन्छ 😂😆	Organization	Organization
Sample 4: @madhuBTM276 पुर्व सचीवको एमाले संग कुनै न कुनै प्रतिशोध छ । त्य सैले एमाले प्रती उहाँ बिष वमन गरिरहनु हुन्छ ।	Individual	Organization

Figure A.4: Few examples of predicted outputs by the proposed method MuRIL-BERT for subtask C