CUET_HateShield@NLU of Devanagari Script Languages 2025: Transformer-Based Hate Speech Detection in Devanagari Script Languages

Sumaiya Rahman Aodhora, Shawly Ahsan and Mohammed Moshiul Hoque
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
{u1804127, u1704057}@student.cuet.ac.bd, moshiul_240@cuet.ac.bd

Abstract

Social networks have become essential platforms for information exchange and free expression. However, their open nature also facilitates the spread of harmful content, such as hate speech, cyberbullying, and offensive language, which pose significant risks to social well-being. This study focuses on developing an automated system to detect hate speech in Devanagari script languages, enabling efficient moderation and timely intervention. Our approach leverages a fine-tuned transformer model for classifying offensive content. We experimented with various machine learning (ML) techniques, including Logistic Regression (LR), Support Vector Machines (SVM), and Random Forest (RF), as well as deep learning (DL) architectures such as CNN, BiLSTM, and CNN-BiLSTM. Additionally, we evaluated transformer-based models, including IndicBERT, m-BERT, MuRIL, Indic-SBERT, and XLM-R. Among these, the fine-tuned XLM-R model delivered the best performance, achieving a macro f_1-score of 0.74, demonstrating its effectiveness in detecting hate speech in Devanagari script languages. However, the model submitted for the shared task achieved a macro f_1-score of 0.73, ranking 13^{th} in the subtask.

1 Introduction

In an increasingly interconnected and digital age, the pervasive impact of communication through social media, online forums, and various digital platforms cannot be overstated. Although these platforms give individuals a voice, they expose them to a spectrum of content, including hate speech. Hate speech, defined as the use of language that disparages or discriminates against individuals or groups based on attributes such as race, ethnicity, religion, gender, politics, or sexual orientation, emerges as a compelling social challenge that requires meticulous attention (Raja Chakravarthi et al., 2021; Parihar et al., 2021). The challenge of manually identifying offensive texts on a large scale emphasizes the urgent need for an automated system to detect and manage hate speech efficiently, enabling faster and more accurate responses to harmful content (Aljarah et al., 2021). The challenge of identifying offensive language has been addressed through various approaches, including the detection of cyberbullying, aggression, toxicity, and abusive language (Sharif et al., 2021; Sharif and Hoque, 2021). However, there is a pressing need for more targeted efforts to specifically address hate speech, especially within diverse linguistic contexts (Singh and Thakur, 2024).

In recent years, significant research efforts have focused on detecting hate and offensive content in high-resource languages like English, Spanish, and Arabic. These benefit from abundant linguistic resources, extensive datasets, and advanced tools (Kumar and Singh, 2022; Omar et al., 2020). However, effectively tackling this issue in low-resource languages remains a significant challenge. To address this challenge, a shared task (Thapa et al., 2025; Sarveswaran et al., 2025) was organized to detect hate speech in the Devanagari script, with a specific focus on monolingual sentences in Nepali and Hindi (Jafri et al., 2024; Thapa et al., 2023). The objective was to determine whether a given sentence contains hate speech, highlighting the importance of effective cross-linguistic detection within the Devanagari script (Jafri et al., 2023; Rauniyar et al., 2023). As participants in this shared task, we contributed to developing and evaluating models tailored for this purpose. The primary contributions of our work are summarized as follows:

- We evaluated various models for hate speech detection, encompassing ML, DL, and transformer-based frameworks, with performance improvements achieved through hyperparameter optimization.
- We conducted a comprehensive comparison of various models, followed by an in-depth

Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), pages 260–266
January 19, 2025.

performance analysis, which led to the proposal of an optimal system for effective hate speech detection.

2 Related Work

In the rapidly advancing field of hate speech detection, researchers have experimented with a wide range of approaches, each playing a role in the ongoing improvement and sophistication of detection models (Parihar et al., 2021). Hate speech detection in Devanagari-script languages, such as Hindi and Nepali, is a technical challenge influenced by social, cultural, and linguistic factors (Parihar et al., 2021). The interpretation of hate speech can vary significantly based on cultural norms, regional dialects, and the social context in which language is used. For instance, certain offensive expressions in one community may not be perceived as such in another (Singh and Thakur, 2024; Thapa et al., 2025). Additionally, the widespread use of code-mixing and social media-specific slang further complicates detecting hate speech. As the field evolved, there was a clear shift from traditional ML techniques to DL, as demonstrated by Omar et al. (2020) in their work on Arabic hate speech detection. They utilized Recurrent Neural Networks (RNN) to achieve a remarkable 98.7% accuracy, outperforming Convolutional Neural Networks (CNN). Sharif and Hoque (2021) employed a weighted ensemble approach combining m-BERT, Distil-BERT, and Bangla-BERT, showcasing the flexibility of these models in capturing complex linguistic variations, especially in Bengali aggressive text datasets.

Shukla et al. (2022) developed a BERT-CNN model for detecting hate speech in low-resource Hindi text, achieving an f_1-score of 0.84. Sharif et al. (2021) tackled the challenge of detecting offensive content in code-mixed social media data by leveraging powerful transformer models such as XLM-R, m-BERT, and Indic-BERT for languages like Tamil, Kannada, and Malayalam. Rauniyar et al. (2023) introduced the NAET dataset, consisting of 4,445 Nepali tweets focusing on political discourse. Their study found that NepNewsBERT outperformed traditional models, achieving an f_1-score of 0.64 in detecting hate speech. Jafri et al. (2024) developed the CHUNAV dataset, which contains Hindi election tweets for hate speech detection and target identification in low-resource languages. They also developed benchmark models, including the Hard Ensemble of BERTs (HEB), demonstrating effective performance with an f_1-score of 0.959.

3 Task and Dataset Description

In this shared task[1], a dataset (Thapa et al., 2025) in Devanagari script containing monolingual Nepali and Hindi sentences was provided to facilitate hate speech detection (Jafri et al., 2024; Thapa et al., 2023). The dataset, designed for binary classification tasks, includes a diverse collection of social media posts and comments, categorized as either hate or non-hate. Participants were provided training, validation, and test datasets to aid model development, validation, and performance evaluation. The training dataset consists of 19,019 samples, with 16,805 non-hate instances and 2,214 hate instances, highlighting a significant class imbalance. A detailed breakdown of additional insights and statistics of the dataset is provided in Table 1.

Classes	Train	Valid	Test	W_T	U_T
Non-Hate	16805	3602	3601	368180	71654
Hate	2214	474	475	58333	19707
Total	19019	4076	4076	426513	91361

Table 1: Class-wise distribution of training, validation, and test sets, where W_T denotes total words and U_T denotes total unique words in the training set

4 Methodology

Figure 1 provides a diagrammatic representation of the approach. We employed various ML and DL techniques to develop the baseline models. Furthermore, we employed five pre-trained transformer models for hate speech detection, including MuRIL, XLM-R, m-BERT, Indic-BERT, and Indic-SBERT.

4.1 Preprocessing

The dataset sourced from social media is characterized by a substantial presence of irrelevant content, including code-mixed elements. Throughout the preprocessing process, we diligently eliminated noise, which comprised hyperlinks, emojis, punctuation, alphanumeric characters, and special symbols (like slashes, brackets, and ampersands) to ensure a higher data quality.

[1] https://codalab.lisn.upsaclay.fr/competitions/20000

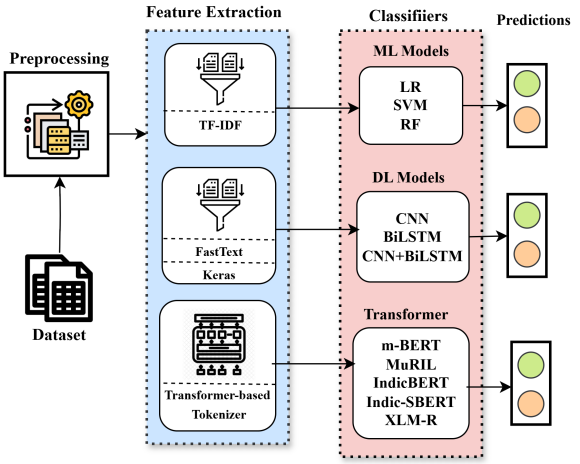

Figure 1: An abstract framework for hate speech detection

4.1.1 Feature Extraction

We applied the TF-IDF technique to extract unigram features for the ML models. TF-IDF assigns weights to words based on their frequency within a document and across the corpus, aiding in identifying significant words that distinguish documents. We employed Keras[2] and pre-trained FastText embeddings for the DL models. FastText embeddings provide 300-dimensional word vectors incorporating subword information through n-grams (Bojanowski et al., 2017; Joulin et al., 2016). Each transformer model utilized its specific tokenizer, obtained from the HuggingFace[3] library, to appropriately tokenize and pad the texts.

4.2 ML Models

In the implementation of LR, the '*lbfgs*' solver was employed alongside balanced class weights and *L2* regularization, with the *C* parameter fine-tuned to mitigate overfitting. The SVM model utilized a *RBF* kernel, with the gamma parameter set to '*scale*' to ensure optimal feature responsiveness. For the RF model, the number of estimators ('*n_estimators*') was configured to 100.

4.3 DL Models

To leverage the effectiveness of DL methods for sequential data analysis, we implemented three approaches: CNN (LeCun et al., 2015), BiLSTM (Hochreiter and Schmidhuber, 1997), and CNN+BiLSTM. The CNN model uses an embedding layer with 256-dimensional embedding, followed by a 1D convolutional layer with 128 filters and a kernel size of 5, concluding with a sigmoid output for binary classification. The BiLSTM model uses an embedding layer with a 300-dimensional embedding size and a maximum sequence length of 100. It then processes the input text through two bidirectional LSTM layers with 64 and 32 units, followed by dropout layers with a rate of 0.5. In the CNN+BiLSTM model, a CNN layer with 128 filters and max-pooling is applied, followed by a 200-cell BiLSTM layer with a dropout rate of 0.2, culminating in final predictions through a sigmoid layer. The hyperparameters for the DL models are shown in Table 2.

Hyperparameters	CNN	BiLSTM	CNN+BiLSTM
Optimizer	Adam	Adam	Adam
Batch Size	32	32	32
Neurons in Dense Layer	64	128	256
Embedding Dimension	256	300	256
Epochs	20	30	30
MaxLen	300	300	300
Dropout Rate	0.2	0.5	0.5
Learning Rate	$1e^{-4}$	$1e^{-3}$	$1e^{-3}$

Table 2: Hyperparameters for DL models

4.4 Transformer Models

We fine-tuned five pre-trained transformer models (MuRIL, XLM-R, m-BERT, Indic-SBERT, and IndicBERT) for hate speech detection in Devanagari script datasets. XLM-R, designed for low-resource languages, uses self-supervised training (Conneau, 2019). Multilingual BERT (m-BERT) was pre-trained on 104 languages (Devlin, 2018), while IndicBERT, covering 12 Indian languages, was trained on a large corpus (Kakwani et al., 2020). Indic-SBERT, a variant of Sentence-BERT, was fine-tuned on a synthetic corpus for Indian languages (Deode et al., 2023). MuRIL, based on BERT, was pre-trained in 17 Indian languages, including transliterated forms and Devanagari scripts (Khanuja et al., 2021). All the transformer models were sourced from the Hugging Face transformer library and fine-tuned on the given dataset using the Ktrain package (Maiya, 2022). The hyperparameters for the transformer-based models are presented in Table 3.

5 Results and Analysis

This section presents a detailed analysis of the effectiveness of various models in detecting hate speech in Devanagari-script languages. The performance of the models is evaluated using the macro

[2] https://keras.io/
[3] https://huggingface.co/

Hyperparameter	m-BERT	MuRIL	IB	ISB	XLM-R
Learning Rate	$2e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$
Batch Size	32	32	16	16	32
MaxLen	100	100	100	100	100
Dropout	0.1	0.1	0.1	0.1	0.1
Epochs	10	10	15	15	10
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW

Table 3: Fine-tuned hyperparameters of the transformer-based models, where IB and ISB represent IndicBERT and Indic-SBERT, respectively.

f_1-score, offering a robust measure of classification accuracy across all classes. Table 4 demonstrates the performance of the employed models.

Approaches	Classifiers	P	R	F1
ML	LR	0.55	0.54	0.54
	SVM	0.53	0.52	0.52
	RF	0.53	0.51	0.51
DL	CNN (FastText)	0.50	0.51	0.50
	BiLSTM (FastText)	0.54	0.52	0.56
	CNN+BiLSTM (FastText)	0.56	0.55	0.55
	CNN (Keras)	0.60	0.59	0.59
	BiLSTM (Keras)	0.62	0.60	0.61
	CNN+BiLSTM (Keras)	0.65	0.62	0.62
Transformer	m-BERT	0.67	0.70	0.69
	MuRIL	0.72	0.73	0.73
	IndicBERT	0.62	0.68	0.64
	Indic-SBERT	0.71	0.75	0.73
	XLM-R	**0.72**	**0.76**	**0.74**

Table 4: Performance of the employed models, where P, R, and F1 denote macro precision, macro recall, and macro f_1-score, respectively

Within the ML category, the LR, SVM, and RF classifiers show competitive performance across precision, recall, and F1 scores, with LR achieving the highest F1 score of 0.54. Those incorporating Keras embeddings for DL models consistently surpass those with FastText embeddings. The top-performing FastText-based model, BiLSTM, reached an F1 score of 0.56. In comparison, the hybrid CNN+BiLSTM model attained an F1 score of 0.62 when using Keras word embeddings. The observed F1 score differences between Keras and FastText embeddings may result from multiple factors. Keras embeddings likely provide more refined contextual representations, capturing linguistic and syntactic patterns that FastText may miss.

In contrast, transformer-based models, especially XLM-R, outperformed both ML and DL models, achieving the highest F1 score of 0.74. MuRIL and Indic-SBERT also performed well, with F1 scores of 0.73. XLM-R's strong performance could be due to its multilingual pretraining, which helps it effectively handle Nepali and Hindi, including the Devanagari script. Moreover, the cross-lingual pretraining of the XLM-R model allowed it to excel despite challenges in the dataset, demonstrating its ability to capture contextual nuances and handle linguistic diversity effectively.

5.1 Classwise Performance

To gain deeper insights, we analyze the best-performing model's classwise performance (XLM-R) as shown in Figure 2. The classification report reveals that the non-hate class has higher precision (0.95) and F1-score (0.93), indicating better performance in identifying non-hate instances. In contrast, the hate class shows a higher recall (0.61), reflecting the ability of the model to accurately identify more true hate instances, though with lower precision (0.49). The comparatively poorer performance in the hate class could be due to the class imbalance.

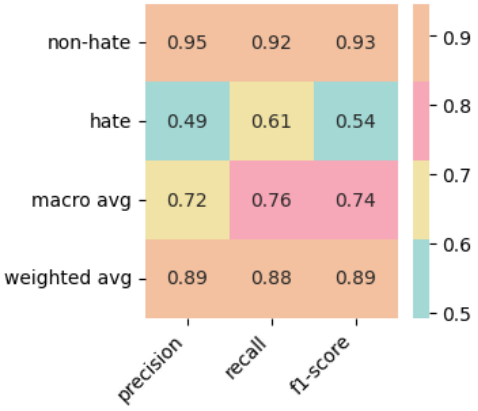

Figure 2: Classwise performance of the best performing model (XLM-R) on the test set.

5.2 Error Analysis

We conducted a comprehensive error analysis using quantitative and qualitative approaches to understand better the performance of the highest-performing model (XLM-R).

5.2.1 Quantitative Analysis

We conducted a quantitative error analysis of the best-performing model (XLM-R) using a confusion matrix (Figure 3). Out of 4,076 samples, 3,592 instances were correctly classified, comprising 3,303 non-hate speech and 289 hate speech samples. However, 484 instances were misclassified, with 186 incorrectly predicted as non-hate and 298 as hate. The higher misclassification rate for hate speech (39.16%) could be attributed to class imbalance, as hate speech samples are significantly fewer. This imbalance hampers the ability

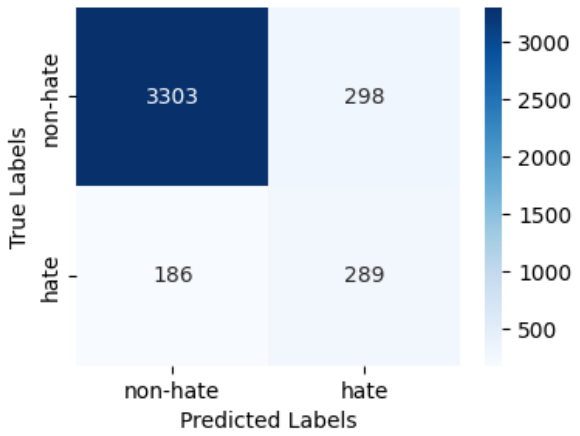

Figure 3: Confusion matrix of the best performing model (XLM-R)

Sample Text	Actual	Predicted
प्रिय कांग्रेसियों, कुमार विश्वास ने इशारों में राहुल गांधी जी को "समलैंगिक" तक कहा है!सोनिया जी के हिंदी उच्चारण का मज़ाक उड़ाया है! उसकी बातों पर सोंच समझकर प्रतिक्रिया दो! (Dear Congressmen, Kumar Vishwas has even called Rahul Gandhi a "gay" in gestures! Made fun of Sonia ji's Hindi pronunciation! Respond thoughtfully to his words!)	non-hate	non-hate
उत्तर प्रदेश में आज छठवें चरण में 10 जनपदों (अम्बेडकरनगर, बलरामपुर, सिद्धार्थनगर, बस्ती, संतकबीर नगर, महराजगंज, गोरखपुर, कुशीनगर, देवरिया, बलिया) की 57 विधानसभा सीटों पर चुनाव चल रहा है। (Today in the sixth phase in Uttar Pradesh, elections are going on for 57 assembly seats in 10 districts (Ambedkarnagar, Balrampur, Siddharthnagar, Basti, Sant Kabir Nagar, Maharajganj, Gorakhpur, Kushinagar, Deoria, Ballia))	non-hate	non-hate
यूक्रेन और रूस का युद्ध समाप्त हो गया क्या 😕🙁 आज तो किसी भी चैनल पर युद्ध की कोई खबर नहीं दिख रही 😁 (Has the war between Ukraine and Russia ended? Today there is no news of war on any channel.)	hate	non-hate
26 फरवरी दिन शनिवार को असदुद्दीन ओवेसी साहब औरपीस पार्टी के राष्ट्रीय अध्यक्ष डॉ अय्यूब साहब (उतरौला) शहर के बरदही बाज़ार में एक विशाल जनसभा को संबोधित करेंगे, इन–शा–अल्लाह उतरौला विधानसभा हम जीत रहे हैं । (On Saturday, 26th February, Asaduddin Owaisi Saheb and National President of Peace Party, Dr. Ayyub Saheb (Utraula) will address a huge public meeting at Bardahi Bazaar of the city, In-Sha-Allah we are winning Utraula assembly.)	hate	non-hate

Table 5: Sample predictions generated by the best-performing model (XLM-R)

of the model to identify hate speech, resulting in increased misclassification accurately.

5.2.2 Qualitative Analysis

Table 5 presents some predicted samples from the best-performing model on the test dataset. Samples 1 and 2 are correctly classified, while samples 3 and 4 are misclassified as non-hate speech, reflecting the model's performance limitations. These misclassifications may result from dataset imbalance, which biases the model toward the majority class, and the presence of code-mixed text complicates language understanding. These challenges underscore the importance of qualitative analysis in interpreting model behavior and identifying areas for improvement.

6 Conclusion

This work contributed to hate speech detection in Devanagari-script languages by systematically evaluating various machine learning (ML), deep learning (DL), and transformer-based models. Among these, the fine-tuned XLM-R model demonstrated the highest performance, achieving a macro f_1-score of 0.74, underscoring the model's capability in effectively classifying offensive content. However, the model exhibited lower performance for the hate speech class, primarily due to class imbalance. Future work will address this issue by employing resampling and data augmentation methods, such as back-translation, to enhance the dataset. Additionally, advanced models, including integrating large language models (LLMs), will be explored to improve performance. Another critical avenue for future research involves developing techniques to effectively handle code-mixed data, particularly Hinglish, to enhance the model's robustness and accuracy.

Limitations

The current approach leverages pre-trained transformer-based models, which, while effective, may need to be revised when the context of the data deviates significantly from the training data. Additionally, due to the lack of specialized mechanisms for handling such linguistic variations, the model's performance could be improved using code-mixed data, such as Hinglish, commonly encountered in Devanagari-script languages. Moreover, the dataset used in this task needed to be more balanced, with certain classes underrepresented. This likely impacted the model's ability to accurately classify instances from these underrepresented classes. Addressing these challenges will be crucial in improving the robustness and accuracy of the model in future work.

References

Ibrahim Aljarah, Maria Habib, Neveen Hijazi, Hossam Faris, Raneem Qaddoura, Bassam Hammo, Mohammad Abushariah, and Mohammad Alfawareh. 2021. Intelligent detection of hate speech in arabic social network: A machine learning approach. *Journal of information science*, 47(4):483–501.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

A Conneau. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Samruddhi Deode, Janhavi Gadre, Aditi Kajale, Ananya Joshi, and Raviraj Joshi. 2023. L3cube-indicsbert: A simple approach for learning cross-lingual sentence representations using multilingual bert. *arXiv preprint arXiv:2304.11434*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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. *arXiv preprint arXiv:2306.14764*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, 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 the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Gunjan Kumar and Jyoti Prakash Singh. 2022. Hate speech and offensive content identification in english and indo-aryan languages using machine learning models. In *FIRE (Working Notes)*, pages 542–551.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436–444.

Arun S Maiya. 2022. ktrain: A low-code library for augmented machine learning. *Journal of Machine Learning Research*, 23(158):1–6.

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. Springer.

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.

Bharathi Raja Chakravarthi, Dhivya Chinnappa, Ruba Priyadharshini, Anand Kumar Madasamy, Sangeetha Sivanesan, Subalalitha Chinnaudayar Navaneethakrishnan, Sajeetha Thavareesan, Dhanalakshmi Vadivel, Rahul Ponnusamy, and Prasanna Kumar Kumaresan. 2021. Developing successful shared tasks on offensive language identification for dravidian languages. *arXiv e-prints*, pages arXiv–2111.

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)*.

Omar Sharif and Mohammed Moshiul Hoque. 2021. Identification and classification of textual aggression in social media: Resource creation and evaluation. In *International Workshop on Combating On line Hostile Posts in Regional Languages dur ing Emergency Situation*, pages 9–20. Springer.

Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2021. Nlp-cuet@ dravidianlangtech-eacl2021: Offensive language detection from multilingual code-mixed text using transformers. *arXiv preprint arXiv:2103.00455*.

Shubham Shukla, Sushama Nagpal, and Sangeeta Sabharwal. 2022. Hate speech detection in hindi language using bert and convolution neural network. In *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pages 642–647. IEEE.

Akshay Singh and Rahul Thakur. 2024. Generalizable multilingual hate speech detection on low resource indian languages using fair selection in federated learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7204–7214.

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.