MDSBots@NLU of Devanagari Script Languages 2025: Detection of Language, Hate Speech, and Targets using MURTweet

Prabhat Ale[*]
IOST, Tribhuvan University
Butwal, Nepal
prabhat.805522@sms.tu.edu.np

Anish Thapaliya[*]
IOST, Tribhuvan University
Kathmandu, Nepal
anish.805522@sms.tu.edu.np

Suman Paudel[*]
IOST, Tribhuvan University
Kathmandu, Nepal
suman.805522@sms.tu.edu.np

Abstract

In multilingual contexts, an automated system for accurate language identification, followed by hate speech detection and target identification, plays a critical role in processing low-resource hate speech data and mitigating its negative impact. This paper presents our approach to the three subtasks in the Shared Task on Natural Language Understanding of Devanagari Script Languages at CHiPSAL@COLING 2025: (i) Language Identification, (ii) Hate Speech Detection, and (iii) Target Identification. Both classical machine learning and multilingual transformer models were explored, where MuRIL Large, trained on undersampled data for subtasks A and B outperformed the classical models. For subtask C, the hybrid model trained on augmented data achieved superior performance over classical and transformer-based approaches. The top-performing models, named MURTweet for subtasks A and B and NERMURTweet for subtask C, secured sixth, third, and first rank respectively, in the competition. The code is publicly available at https://github.com/thapaliya123/CHIPSAL-COLING-2025.

1 Introduction

Social media platforms like Twitter (currently called X) are filled with various types of hate speech, including racism, sexism, hate speech related to religion, and more (Parihar et al., 2021). Advances in large language models for regional languages, like Nepali, help detect hate speech and analyze sentiment on social media platforms (Pudasaini et al., 2024). Political leaders also use Twitter to spread their political agendas by sharing news and information with a broad audience and participating in discussions, either supporting or criticizing political actions (Lai et al., 2023). Politicians and their followers may use phrases, expressions, and words of hate to gain an advantage and belittle other parties (Wang et al., 2022).

The shared work on Natural Language Understanding of Devanagari Script Languages, organized as part of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL) (Sarveswaran et al., 2025), allows participants to investigate approaches for recognizing low-resource South Asian languages, moving beyond the widely used English language, even with domain-specific models (Mali et al., 2024). It also focuses on detecting hate speech and identifying its targets, which are often political individuals, organizations, or specific community groups (Thapa et al., 2025).

The MuRIL Large model[1], a specialized BERT large (24-layer) architecture (Khanuja et al., 2021), has been chosen as a solution to the shared task because of its strong multilingual capabilities, particularly for Devanagari languages. In this method, fine-tuning was applied across all model parameters to calibrate the MuRIL architecture for tasks at hand, including language identification, hate speech detection, and target identification.

The undersampling technique was used to reduce training time for subtask A and to address class imbalance in subtask B. For subtask C, data augmentation techniques were applied to address a class imbalance problem. A synonym replacement technique was used to generate synthetic samples from validation data, resulting in improved model performance. A rule-based Named Entity Recognition (NER) approach was also used to classify individual tokens associated with individuals, organizations, or community groups in tweets, with the goal of identifying hate speech targets for subtask C. F1 score was chosen as the primary metric, as leaderboard rankings were based on the F1 score and also the data were imbalanced. This study provides an overview of the solution's results, high-

[*] *These authors contributed equally to this work.

[1] https://huggingface.co/google/muril-large-cased

Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), pages 308–313
January 19, 2025.

lighting a sixth-place ranking in subtask A, third in subtask B, and first place in subtask C.

2 Subtasks and Datasets

2.1 Subtask A: Language Identification

This work examines five Devanagari-script languages: Nepali, Marathi, Sanskrit, Bhojpuri, and Hindi. To achieve accurate language identification and recognition across these languages, the following datasets are used:

1) Nepali (Thapa et al., 2023; Rauniyar et al., 2023), 2) Marathi (Kulkarni et al., 2021), 3) Sanskrit (Aralikatte et al., 2021), 4) Bhojpuri (Ojha, 2019), 5) Hindi (Jafri et al., 2024, 2023).

2.2 Subtask B: Hate Speech Detection

Subtask B aims to detect hate speech in Nepali and Hindi tweets. The task aims to identify hate speech in Nepali and Hindi tweets, where participants must classify each tweet as either containing hateful statements or not.

2.3 Subtask C: Target Identification

Subtask C aims to identify the targets of hate speech in a given tweet. The dataset for this subtask is annotated for ”individual”, ”organization”, and ”community” targets.

2.4 Datasets

An open-source version of the dataset for all three subtasks is available on the competition's webpage [2]. The distribution of each class within the dataset for each subtask is provided in Table 1.

Table 1: Dataset distribution for each class across Train, Validation, and Test sets for subtasks A, B, and C.

Task	Class	Train	Validation	Test
A	Nepali	12544	2688	2688
	Marathi	11034	2364	2365
	Sanskrit	10996	2356	2356
	Bhojpuri	10184	2182	2183
	Hindi	7664	1643	1642
B	Hate	2214	474	475
	Non-Hate	16805	3602	3601
C	Individual	1074	230	230
	Organization	856	183	184
	Community	284	61	61

3 Methodology

The strategy includes three major solutions. Initially, a classical machine learning strategy was used, with typical models trained on the dataset. Then, transformer-based models were investigated, encompassing both encoder models such as variations of BERT (Devlin, 2018), and decoder models such as GPT (Achiam et al., 2023). Finally, a hybrid strategy was used, in which NER tags were applied to the data using a rule-based tagger, followed by inference on the NER-tagged data using the trained model. Figure 1 shows the block diagram of the proposed model's training and inference pipeline.

3.1 Augmentation and Undersampling

Both augmentation and undersampling procedures were used to solve class imbalance issues and to reduce training time (Pimpalkhute et al., 2021). Subtasks A and B were undersampled, while subtask C was augmented with synthetic data. In subtask A, due to the large number of training samples, the model was trained only on 50% of the overall data to reduce the training and hyperparameter tuning time. In subtask B, the majority class was undersampled, with a focus on samples from the non-hate class to reduce class imbalance. For subtask C, the minority class, representing community targets, was augmented. This approach was chosen because the limited number of community target samples made undersampling the majority class impractical and potentially harmful to performance on the test set. For the community class in subtask C, a synonym substitution technique and GPT-4 in-context learning were used, along with validation samples and prompting, to generate synthetic tweets with identical sentiments (Zhang et al., 2024). Table 2 shows the distribution of training samples before and after applying undersampling or data augmentation techniques. The prompt used to generate synthetic samples through the synonym replacement technique is provided in Figure 2 in Appendix Section A.

3.2 Classical Approach

A classical approach was employed, where features were extracted from textual data using Term Frequency-Inverse Document Frequency (TF-IDF), and multiple machine learning algorithms such as Logistic Regression (LR), Support Vector Machines (SVM), Random Forest (RF), and XGBoost (XGB) were trained to establish a baseline for experiments.

[2] https://sites.google.com/view/chipsal/shared-tasks

Table 2: Train data distribution across subtasks A, B, and C: Undersampling for subtasks A, B and data augmentation for subtask C

Task	Class	Before Sampling /Augmentation	After Sampling /Augmentation
A	Nepali	12544	6231
	Marathi	11034	5535
	Sanskrit	10996	5447
	Bhojpuri	10184	5156
	Hindi	7664	3842
B	Hate	2214	2214
	Non-Hate	16805	8437
C	Individual	1074	1074
	Organization	856	856
	Community	284	330

3.3 Transformer Based Approach

Experiments were carried out employing Transformer architectures (Vaswani, 2017), with a focus on two variants: encoder-only models and decoder-only models.

3.3.1 Encoder Models

Encoder-only models, such as BERT variants (Devlin, 2018), utilize Transformer architectures' encoder layers. The encoder models listed below have been tested in all subtasks and optimized for low-resource Devanagari languages.

mBERT: BERT (Bidirectional Encoder Representation from Transformers) (Devlin, 2018) is self-trained using Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) on BookCorpus and Wikipedia. The multilingual extension of BERT, mBERT, is trained in 104 languages and is offered in two versions: BERT-base and BERT-large.

XLM-RoBERTa: XLM-RoBERTa(Conneau, 2019) is a Transformer-based masked language model trained on over two terabytes of filtered CommonCrawl data from 100 languages. XLM-RoBERTa is specifically developed for low-resource languages.

Varta - A Large-Scale Headline-Generation Dataset for Indic Languages (Aralikatte et al., 2023): Varta-BERT is a model pre-trained on the entire Varta dataset, which includes 14 Indic languages along with English. The model is trained with the masked language modeling (MLM) objective.

MuRIL: Multilingual Representations for Indian Languages (Khanuja et al., 2021) is a BERT-based architecture that was pre-trained from scratch using data from Wikipedia, Common Crawl, PMINDIA, and Dakshina corpora in 17 Indian languages. There are two variants of the model available: base and large, with pre-trained weights available on Hugging Face.

3.3.2 Decoder Models

Prompt Engineering and Few-Shot Learning have become prominent methods for detecting hate speech on Twitter (Dehghan and Yanikoglu, 2024). This work utilizes decoder-only models developed by OpenAI (gpt-4-2024-05-13)[3], which excel at multilingual tasks (Achiam et al., 2023). Since OpenAI models are not open-source, they were accessed via API endpoints. Due to competition time constraints, experiments were conducted only with OpenAI models for subtask C. Validation splits were used for prompt tuning, while few-shot learning employed samples from the training splits. To achieve consistent JSON outputs, the temperature parameter was set to zero, reducing the non-deterministic responses typical of large language models. After tuning, the final prompt is provided in Figure 3 in Appendix Section B.

3.3.3 Hybrid Models

Word knowledge has been obtained by extracting entity information from Wikipedia and feeding it into the model alongside hate speech text(Lin, 2022). (Kaya et al., 2024) has employed a hybrid approach that combines: (1) reclassifying samples with low confidence scores using open-source large language models via prompting, and (2) integrating named entity information into features generated by BERT models, with the final output produced through tree-based models.

A hybrid approach has been used for subtask C, starting with error analysis on low-confidence samples (probability < 0.6) and re-evaluating them after adding entity tags. A rule-based entity tagger applied four tags व्यक्ति(person), संगठन(organization), चुनाव चिन्ह (election symbol), and समूह(group) around relevant phrases, using predefined entity lists for each category. These tags were selected after extensive experimentation to closely align with the given hate speech targets. Since the dataset was more directly tied to political news, election symbols were included in the tag pool because they resemble political organizations. The tagged samples were then passed through the trained MURTweet model, resulting in an enhanced version named NERMURTweet (see

[3] https://platform.openai.com/docs/models/o1#gpt-4o

Figure 1), which significantly improved classification metrics. Examples of samples before and after entity tagging are shown in Figure 4 in Appendix section C.

The suggested approach used a batch size of 32 for subtask A and 8 for subtasks B and C. The learning rate was set to 2e-5 with the AdamW optimizer, weight decay of 0.001, and trained for 10 epochs.

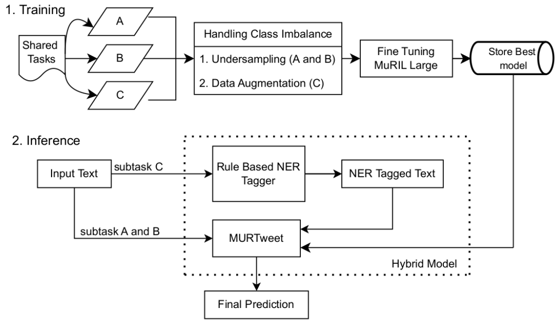


Figure 1: Block Diagram For training and inference pipeline

3.4 Results And Discussions

The table 3 displays the leaderboard results for all subtasks in the experiment. Precision, recall, F1 score, and accuracy metrics are presented, and models are ordered according to the F1 score specified by the organizers. For each challenge, the models with the highest F1 score are shown in bold.

In subtask A, the proposed model, trained on 50% of the total data, has secured sixth position in the final ranking. In subtask B, the proposed model, trained using undersampled data, secured the third position in the final scoreboard. For subtask C, the hybrid model (MURTweet + NER Tagging) secured first position in the final ranking. Due to the large volume of data, subtask A has been trained with only half of the dataset. MURTweet has the highest F1 score of 0.9962, surpassing Vartha, XLM-RoBERTa, and mBERT. In subtask B, undersampling the majority class in the training data enhanced model performance. The MURTweet model trained on undersampled data outperforms the model trained on the entire dataset. For subtask C, a hybrid model combining MURTweet with rule-based NER tagging topped the leaderboard, along with error analysis-driven data augmentation approaches (such as synonym matching and paraphrasing) improving performance. Attempts to employ GPT-4 for target identification in subtask C produced an F1 score of 0.66, which was not higher than the hybrid-based approach.

Table 3: Performance metrics across subtasks A, B, and C.

Task	Model	Prec	Rec	F1	Acc
A	TFIDF + LR	0.9528	0.9521	0.9526	0.9542
	TFIDF + SVM	0.9627	0.9626	0.9636	0.9662
	TFIDF + RF	0.9799	0.9776	0.9786	0.9806
	TFIDF + XGB	0.9439	0.9446	0.9442	0.9475
	mBERT	0.9930	0.9930	0.9930	0.9936
	XLM-R-Base	0.9918	0.9927	0.9922	0.9931
	XLM-R-Large	0.9940	0.9942	0.9941	0.9947
	Varta-BERT	0.9952	0.9957	0.9954	0.9959
	MuRIL-Base	0.9948	0.9953	0.9950	0.9955
	MURTweet	**0.9967**	**0.9968**	**0.9968**	**0.9972**
B	TFIDF + LR	0.5759	0.6843	0.5020	0.5746
	TFIDF + SVM	0.5700	0.6699	0.4894	0.5589
	TFIDF + RF	0.5597	0.5108	0.4987	0.8734
	TFIDF + XGB	0.6721	0.5345	0.5377	0.8815
	mBERT	0.6336	0.7136	0.6542	0.8121
	XLM-R-Base	0.6685	0.7147	0.6867	0.8528
	XLM-R-Large	0.7068	0.7542	0.7264	0.8741
	Varta-BERT	0.6179	0.7037	0.6344	0.7897
	MuRIL-Base	0.6803	0.7715	0.7089	0.8481
	MURTweet	**0.7638**	**0.7687**	**0.7662**	**0.9028**
C	TFIDF + LR	0.5169	0.5223	0.5147	0.5811
	TFIDF + SVM	0.4103	0.4107	0.4101	0.5705
	TFIDF + RF	0.4287	0.4414	0.4315	0.5747
	TFIDF + XGB	0.4641	0.4641	0.4641	0.5579
	mBERT	0.6022	0.6059	0.6036	0.6780
	XLM-R-Base	0.6319	0.6365	0.6339	0.7034
	XLM-R-Large	0.7095	0.6891	0.6975	0.7648
	Varta-BERT	0.6751	0.6410	0.6514	0.7331
	MuRIL-Base	0.6450	0.6576	0.6500	0.70763
	MURTweet	0.7073	0.6867	0.6951	0.7621
	NERMURTweet	**0.7175**	**0.7038**	**0.7098**	**0.7684**

4 Limitations

For subtasks A and B, undersampled data has been used for training, so the full potential of the complete dataset has not been fully utilized. In subtask C, the rule-based NER tagger assigned tags to phrases that do not represent real-world entities, which could have affected the final prediction.

5 Conclusion

In conclusion, data augmentation and undersampling techniques have shown impressive results in addressing class imbalance problems through this research. Encoder-only models trained on undersampled data have outperformed traditional methods in language recognition and hate speech detection tasks. For target identification, results have shown that hybrid models, which used the best-performing models on NER-tagged data, outperformed encoder-only and generative models, highlighting the importance of NER information in successfully identifying hate speech targets. Future research should explore ML-based NER tagging approaches, as well as alternative class imbalance techniques like weighted cross-entropy loss and focal loss, to further enhance model performance in these tasks.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rahul Aralikatte, Ziling Cheng, Sumanth Doddapaneni, and Jackie Chi Kit Cheung. 2023. V\= arta: A large-scale headline-generation dataset for indic languages. *arXiv preprint arXiv:2305.05858*.

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.

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

Somaiyeh Dehghan and Berrin Yanikoglu. 2024. Evaluating chatgpt's ability to detect hate speech in turkish tweets. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*, pages 54–59.

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

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.

Ahmet Kaya, Oguzhan Ozcelik, and Cagri Toraman. 2024. Arc-nlp at climateactivism 2024: Stance and hate speech detection by generative and encoder models optimized with tweet-specific elements. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*, pages 111–117.

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*.

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.

Mirko Lai, Fabio Celli, Alan Ramponi, Sara Tonelli, Cristina Bosco, and Viviana Patti. 2023. Haspeede3 at evalita 2023: Overview of the political and religious hate speech detection task.

Jessica Lin. 2022. Leveraging world knowledge in implicit hate speech detection. *arXiv preprint arXiv:2212.14100*.

Drish Mali, Rubash Mali, and Claire Barale. 2024. Information extraction for planning court cases. In *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 97–114.

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

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.

Varad Pimpalkhute, Prajwal Nakhate, and Tausif Diwan. 2021. Iiitn nlp at smm4h 2021 tasks: transformer models for classification on health-related imbalanced twitter datasets. In *Proceedings of the Sixth Social Media Mining for Health (# SMM4H) Workshop and Shared Task*, pages 118–122.

Shushanta Pudasaini, Sunil Ghimire, Prabhat Ale, Aman Shakya, Prakriti Paudel, and Basanta Joshi. 2024. Application of nepali large language models to improve sentiment analysis. In *Proceedings of the 2024 7th International Conference on Computers in Management and Business*, pages 144–150.

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)*.

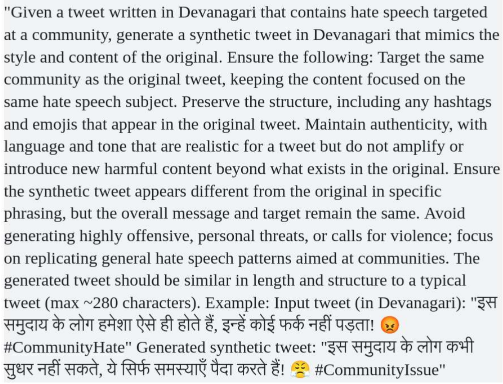
Surendrabikram Thapa, Kritesh Rauniyar, Shuvam Shiwakoti, Sweta Poudel, Usman Naseem, and Mehwish Nasim. 2023. Nehate: Large-scale annotated data shedding light on hate speech in nepali local election discourse. In *ECAI 2023*, pages 2346–2353. IOS Press.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Chih-Chien Wang, Min-Yuh Day, and Chun-Lian Wu. 2022. Political hate speech detection and lexicon building: A study in taiwan. *IEEE Access*, 10:44337–44346.

Yaqi Zhang, Viktor Hangya, and Alexander Fraser. 2024. A study of the class imbalance problem in abusive language detection. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pages 38–51.

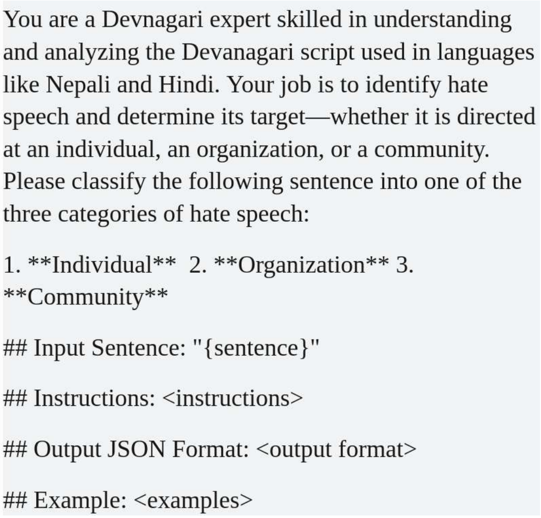
A Appendix: Prompt for Synthetic Sample Generation



"Given a tweet written in Devanagari that contains hate speech targeted at a community, generate a synthetic tweet in Devanagari that mimics the style and content of the original. Ensure the following: Target the same community as the original tweet, keeping the content focused on the same hate speech subject. Preserve the structure, including any hashtags and emojis that appear in the original tweet. Maintain authenticity, with language and tone that are realistic for a tweet but do not amplify or introduce new harmful content beyond what exists in the original. Ensure the synthetic tweet appears different from the original in specific phrasing, but the overall message and target remain the same. Avoid generating highly offensive, personal threats, or calls for violence; focus on replicating general hate speech patterns aimed at communities. The generated tweet should be similar in length and structure to a typical tweet (max ~280 characters). Example: Input tweet (in Devanagari): "इस समुदाय के लोग हमेशा ऐसे ही होते हैं, इन्हें कोई फर्क नहीं पड़ता! 😡 #CommunityHate" Generated synthetic tweet: "इस समुदाय के लोग कभी सुधर नहीं सकते, ये सिर्फ समस्याएँ पैदा करते हैं! 😤 #CommunityIssue"

Figure 2: Prompt for Synthetic Sample Generation

B Appendix: Prompt for Hate Speech Target Identification



You are a Devnagari expert skilled in understanding and analyzing the Devanagari script used in languages like Nepali and Hindi. Your job is to identify hate speech and determine its target—whether it is directed at an individual, an organization, or a community. Please classify the following sentence into one of the three categories of hate speech:

1. **Individual** 2. **Organization** 3. **Community**

Input Sentence: "{sentence}"

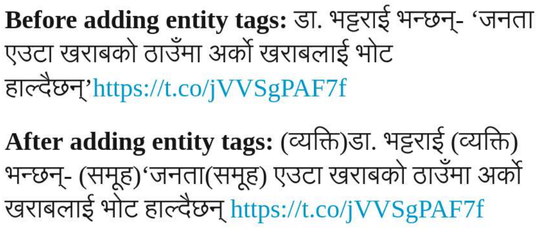
Instructions: <instructions>

Output JSON Format: <output format>

Example: <examples>

Figure 3: Prompt for Synthetic Sample Generation

C Appendix: Impact of NER Tagging on Hate Speech Tweets



Before adding entity tags: डा. भट्टराई भन्छन्- ‘जनता एउटा खराबको ठाउँमा अर्को खराबलाई भोट हाल्दैछन्’https://t.co/jVVSgPAF7f

After adding entity tags: (व्यक्ति)डा. भट्टराई (व्यक्ति) भन्छन्- (समूह)‘जनता(समूह) एउटा खराबको ठाउँमा अर्को खराबलाई भोट हाल्दैछन् https://t.co/jVVSgPAF7f

Figure 4: Impact of NER Tagging on Hate Speech Tweets