Dll5143A@NLU of Devanagari Script Languages 2025: Detection of Hate Speech and Targets Using Hierarchical Attention Network

Ashok Yadav and **Vrijendra Singh**
Indian Institute of Information Technology Allahabad
Prayagraj, 211015, India
rsi2021002@iiita.ac.in, vrij@iiita.ac.in

Abstract

Hate speech poses a significant challenge on social networks, particularly in Devanagari scripted languages, where subtle expressions can lead to harmful narratives. This paper details our participation in the "Shared Task on Natural Language Understanding of Devanagari Script Languages" at CHIPSAL@COLING 2025, addressing hate speech detection and target identification. In Sub-task B, we focused on classifying the text either hate or non-hate classified text to determine the presence of hate speech, while Sub-task C focused on identifying targets, such as individuals, organizations, or communities. We utilized the XLM-RoBERTa model as our base and explored various adaptations, including Adaptive Weighting and Gated Adaptive Weighting methods. Our results demonstrated that the Hierarchical Gated adaptive weighting model achieved 86% accuracy in hate speech detection with a macro F1 score of 0.72, particularly improving performance for minority class detection. For target detection, the same model achieved 75% accuracy and a 0.69 macro F1 score. Our proposed architecture demonstrated competitive performance, ranking 8th in Subtask B and 11th in Subtask C among all participants.

1 Introduction

In the age of rapid digital communication, social media platforms have become the primary space for people to share their opinions and engage in discourse (Zhou et al., 2024). However, this democratization of speech has also led to the propagation of hate speech, which can have severe consequences for individuals and communities (Parida et al., 2024). While hate speech detection in major languages like English has seen significant advancements, there is a pressing need to extend this effort to languages written in Devanagari scripts, such as Nepali, Marathi, Sanskrit, Bhojpuri, and Hindi (Rauniyar et al., 2023). These languages, despite their widespread use in South Asia, remain underrepresented in hate speech research (Piot et al., 2024). Devanagari script languages present unique challenges for hate speech detection due to their linguistic structure, rich cultural context, and scarcity of labeled datasets (Parihar et al., 2021). Existing hate speech detection models, predominantly trained in English or other high-resource languages, often fail to capture the nuances of these languages, leading to poor performance. Furthermore, the intertwining of hate speech with regional socio-political issues adds layers of complexity that existing models are not equipped to handle (Jafri et al., 2024).

To address these challenges, (Sarveswaran et al., 2025) introduced a shared task at CHIPSAL@COLING 2025 where participants' systems need to detect the language (Nepali, Marathi, Sanskrit, Bhojpuri, or Hindi) a given Devanagari text belongs to, as well as identify hate speech and its targets within the text. Subtask B of the challenge contains tweets that were carefully classified into two groups, hate and non-hate. Subtask C focuses on identifying whether the hate speech targets 'Individual', 'Organization', or 'Community', allowing for detailed tracking of hate speech communication patterns in Devanagari. We have participated in Subtask B and Subtask C, and achieved ranks 8th and 11th respectively, among all participants.

The remainder of this paper is organized as follows: Section 2 introduces the shared task and dataset statistics. Section 3 details our hierarchical attention-based architecture. Section 4 describes the experimental setup and evaluation metrics. Section 4.1 presents our model's performance and comparison with other participating systems. Section 5 concludes our findings, while Section 6 discusses the limitations and potential improvements in hate speech detection for Devanagari script languages.

2 Task and Dataset Description

The 'Shared Task on Natural Language Understanding of Devanagari Script Languages' at CHIPSAL@COLING 2025 focuses on key challenges in processing Devanagari-scripted languages. The first subtask, Devanagari Script Language Identification, aims to accurately identify the language of a given Devanagari text. Subtask B, Hate Speech Detection, determines whether a text contains hate speech. Building on this, Subtask C targets identifying specific hate speech targets, such as individuals or groups. This shared task promotes comprehensive Devanagari language understanding by addressing script identification, hate speech detection, and hate speech target identification. (Thapa et al., 2025).

The hate speech datasets for this shared task were drawn from various sources. For Hindi, the CHUNAV dataset (Jafri et al., 2024) and a dataset on political hate speech during Indian elections (Jafri et al., 2023) were used. The Nepali dataset, NEHATE (Thapa et al., 2023), and a multi-aspect dataset on Nepali tweets regarding anti-establishment election discourse (Rauniyar et al., 2023) were also included. Datasets for Bhojpuri (Ojha, 2019), Marathi (Kulkarni et al., 2021), and Sanskrit (Aralikatte et al., 2021) were utilized. In this shared task, we participated in both Subtask B and Subtask C, achieving ranks of 8th and 11th respectively among all participants. The dataset for Subtask B includes binary annotations (0 for non-hate and 1 for hate speech), while Subtask C focuses on categorizing hate speech targets into three classes: individual (0), organization (1), and community (2). Table 1 provides statistics on the dataset used for the ChiPSAL shared task for both subtasks.

Table 1: The statistics of the used dataset in CHIPSAL@COLING 2025 Subtask B and Subtask C.

Category	Subask B		Subtask C		
	Hate	Non-Hate	Individual	Organization	Community
Train	2214	1,6805	1074	856	284
Val	474	3602	230	183	61
Test	475	3601	230	184	61
Total	3164	24008	1534	1223	406

3 Proposed Framework

3.1 Overview

We have proposed a model that builds on the XLM-RoBERTa architecture, incorporating adaptive attention mechanisms to improve classification performance in diverse linguistic contexts. Figure 1 shows the architecture of our proposed model.

3.2 XLM-RoBERTa

XLM-RoBERTa serves as the foundation of our model. It is a multilingual transformer with 12 layers, 768 hidden units, and 12 attention heads. This base model processes the input text and generates contextualized word embeddings, also known as hidden states. These hidden states, denoted as $H \in \mathcal{R}^{B\times L\times D}$, where B is the batch size, L is the sequence length, and D is the hidden size (768), serve as the features extracted from the input text and form the basis for subsequent processing in our model.

3.3 Attention Mechanism

We implement a dual-attention mechanism consisting of word-level and sentence-level attention components.

3.3.1 Word-level Attention

The word-level attention component is a two-layer feedforward neural network that processes the hidden states to generate attention weights for individual tokens. The process can be described by the following equations:

$$e_w = \tanh(W_w^1 H + b_w^1) \tag{1}$$

$$\alpha_w = \text{softmax}(W_w^2 e_w + b_w^2) \tag{2}$$

$$c_w = \sum_{i=1}^{L} \alpha_w^i H^i \tag{3}$$

where $W_w^1 \in \mathcal{R}^{D\times D}$, $W_w^2 \in \mathcal{R}^{1\times D}$, $b_w^1 \in \mathcal{R}^D$, and $b_w^2 \in \mathcal{R}$ are learnable parameters, $\alpha_w \in \mathcal{R}^L$ are the attention weights, and $c_w \in \mathcal{R}^D$ is the word-level context vector.

3.3.2 Sentence-level Attention

The sentence-level attention mechanism focuses on broader semantic structures within the input. It follows a similar structure to the word-level attention:

$$e_s = \tanh(W_s^1 H + b_s^1) \tag{4}$$

$$\alpha_s = \text{softmax}(W_s^2 e_s + b_s^2) \tag{5}$$

$$c_s = \sum_{i=1}^{L} \alpha_s^i H^i \tag{6}$$

where W_s^1, W_s^2, b_s^1, and b_s^2 are learnable parameters with the same dimensions as their word-level counterparts, $\alpha_s \in \mathcal{R}^L$ are the sentence-level attention weights, and $c_s \in \mathcal{R}^D$ is the sentence-level context vector.

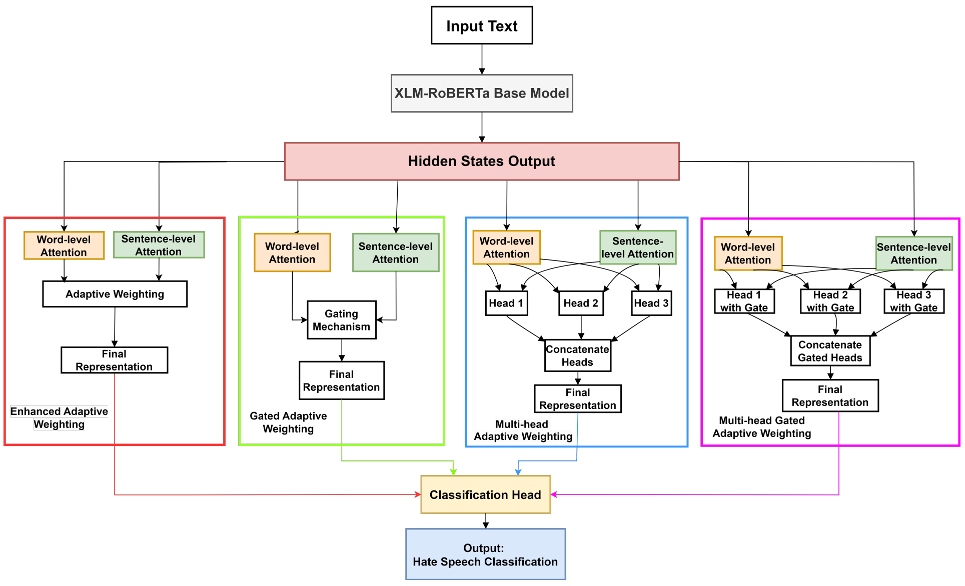


Figure 1: Architecture of the proposed transformer-based multimodal hierarchical fusion model.

3.3.3 Adaptive Weighting

The adaptive weighting component combines the word-level and sentence-level context vectors based on their relative importance for each input.

$$c_{combined} = [c_w; c_s] \tag{7}$$

$$\beta = \text{softmax}(W_a^2 \text{ReLU}(W_a^1 c_{combined} + b_a^1) + b_a^2) \tag{8}$$

$$c_{final} = \beta_1 c_w + \beta_2 c_s \tag{9}$$

where $W_a^1 \in \mathcal{R}^{D \times 2D}$, $W_a^2 \in \mathcal{R}^{2 \times D}$, $b_a^1 \in \mathcal{R}^D$, and $b_a^2 \in \mathcal{R}^2$ are learnable parameters, $\beta \in \mathcal{R}^2$ are the adaptive weights, and $c_{final} \in \mathcal{R}^D$ is the final context vector that balances word and sentence-level information.

We also explored advanced variants of the attention mechanism including gated adaptive weighting, multi-head adaptive weighting, and multi-head gated adaptive weighting. The detailed architectures and formulations of these variants are presented in Appendix 7.

3.4 Classification Head

The classification head is a three-layer MLP with ReLU activations and dropout that takes the weighted representation and outputs logits for both two-way and three-way classifications. The forward pass follows:

$$h_1 = \text{ReLU}(W_1 c_{final} + b_1) \tag{10}$$

$$h_2 = \text{ReLU}(W_2 h_1 + b_2) \tag{11}$$

$$z = W_3 h_2 + b_3 \tag{12}$$

where $W_1 \in \mathcal{R}^{D \times D}$, $W_2 \in \mathcal{R}^{D/2 \times D}$, $W_3 \in \mathcal{R}^{2 \times D/2}$, and corresponding biases are learnable parameters. To handle class imbalance, we use weighted Cross-Entropy Loss:

$$\mathcal{L} = -\sum_{i=1}^{N} w_{y_i} [y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))] \tag{13}$$

with class weights calculated using sklearn's 'balanced' strategy:

$$w_j = \frac{N}{K \cdot N_j} \tag{14}$$

where N is total samples, K is number of classes, and N_j is samples in class j.

4 Experimental Settings and Evaluations Metrics

We implemented our model using PyTorch and Hugging Face Transformers, with training conducted on an Nvidia A30 GPU. The model was trained for 10 epochs using an AdamW optimizer with a learning rate of 2e-5 and batch size of 32. For reproducibility, we set the manual seed to 30 and used a dropout rate of 0.3 to prevent overfitting. Input sequences were padded to a maximum length of 128 tokens. In our implementation, all layers of XLM-RoBERTa were fine-tuned during training to maximize its full representational capacity for contextual and linguistic understanding. The system's effectiveness was assessed using standard metrics: precision, recall, F1-score, and accuracy.

The macro-averaged F1 score was selected as the primary evaluation criterion for both subtasks.

4.1 Results and Analysis

On the validation dataset, our best performing model, the Hierarchical Gated Adaptive Attention model, achieved an F1 score of 0.72, precision of 0.70, recall of 0.76, and accuracy of 0.86 for hate speech detection (Subtask B). For target identification (Subtask C), the model attained an F1 score of 0.69, precision of 0.69, recall of 0.69, and accuracy of 0.75. Tables 2 and 3 present our system's performance compared to other participating systems on the test dataset.

For hate speech detection (Subtask B) [1], our system achieved competitive results, ranking 8th among all participants with an F1 score of 0.745 and an accuracy of 0.890. The best performing system achieved an F1 score of 0.814, demonstrating the challenging nature of hate speech detection in Devanagari script languages. Our model showed balanced performance between precision (0.735) and recall (0.758), indicating its effectiveness in handling class imbalance. In target identification (Subtask C) [2], our system ranked 11th with an F1 score of 0.658 and accuracy of 0.714. While the top system achieved an F1 score of 0.710, the relatively small performance gap (0.052) between the first and eleventh positions suggests the complexity of the task and the effectiveness of various approaches.

Table 2: Results comparison of top systems for Subtask B, R(recall),P(precision),and Acc (accuracy)

System	R	P	F1	Acc	Rank
fulbutte	.855	.785	.814	.914	1
Yestin	.813	.746	.773	.894	2
sumanpaudel	.769	.763	.766	.903	3
jebish7	.744	.793	.765	.911	4
lazyboy.blk	.736	.790	.759	.910	5
MuhammadA	.726	.781	.749	.907	6
mdp0999	.775	.729	.748	.886	7
Ours	.758	.735	.745	.890	8

Detailed performance analysis of all model variants of subtask B 8.1 and Subtask C 8.2 is presented in Appendix 8.

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

[2]https://codalab.lisn.upsaclay.fr/competitions/20000#results

Table 3: Results comparison of top systems for Subtask C

System	R	P	F1	Acc	Rank
sumanpaudel	.704	.718	.710	.768	1
Siddartha-10	.687	.741	.703	.779	2
Tofa	.672	.742	.692	.766	3
sakib07	.681	.686	.683	.745	4
Dola_C	.681	.679	.680	.737	5
jebish7	.669	.697	.679	.750	6
mdp0999	.669	.674	.672	.741	7
jerrytomy	.667	.663	.664	.731	8
sandeep_S	.657	.675	.664	.739	9
Yestin	.655	.674	.661	.745	10
Ours	.654	.664	.658	.714	11

5 Conclusion

In this study, we explored hate speech detection and target identification challenges in Devanagari-scripted languages through our participation in CHIPSAL@COLING 2025. Our experimentation with various attention mechanisms demonstrated that the Hierarchical Gated Adaptive Weighting model achieved the best performance, with macro F1 scores of 0.72 and 0.69 for hate speech detection and target identification respectively. The integration of gating mechanisms proved crucial in addressing class imbalance, particularly improving minority class detection in both tasks. Despite achieving competitive rankings—8th in Subtask B with an F1 score of 0.745 and 11th in Subtask C with an F1 score of 0.658—our analysis revealed persistent challenges. The model showed stronger performance in detecting individual (F1: 0.79) and organizational targets (F1: 0.77) but struggled with community-targeted hate speech (F1: 0.52), highlighting the complexity of detecting group-targeted hate. This performance disparity suggests the need for more sophisticated approaches to handle the nuanced expressions of community-targeted hate speech in Devanagari languages.

6 Limitations

Our work contributes to the research on processing low-resource languages, demonstrating how hierarchical attention models with adaptive weighting can significantly enhance performance. However, the model struggles to detect community-targeted hate speech (F1: 0.52) compared to individual (F1: 0.79) and organizational targets (F1: 0.77). This performance gap highlights the model's difficulty in handling unbalanced data for target detection, especially in recognizing hate speech directed at specific communities. Community-targeted tweets often employ indirect or culturally nuanced language, as detailed in Appendix 9.2. We used XLM-RoBERTa as our base model, which, despite its robust multilingual capabilities, may lack the nuanced script-specific features required for Devanagari. This limitation is particularly evident in handling code-mixed language or symbolic terms. We observed that named entities and metaphorical phrases—common in political discourse—were frequently misinterpreted, leading to false positives in hate speech detection. Detailed examples of these challenges can be found in Appendix 9.1. To address these limitations, future work could include specialized pre-training methods that better handle linguistic and cultural elements inherent in Devanagari languages. Exploring script-specific models or training strategies may also help the model distinguish between satirical and hateful language more effectively, especially in community-oriented contexts where expression style differs significantly.

Acknowledgments

The authors express their gratitude to the Ministry of Education, the Indian Institute of Information Technology Allahabad, and the Deep Learning Lab at IIITA for providing the resources necessary to complete this work.

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.

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.

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.

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

Shantipriya Parida, Shakshi Panwar, Kusum Lata, Sanskruti Mishra, and Sambit Sekhar. 2024. Building pre-train llm dataset for the indic languages: a case study on hindi. *arXiv preprint arXiv:2407.09855*.

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.

Paloma Piot, Patricia Martín-Rodilla, and Javier Parapar. 2024. Metahate: A dataset for unifying efforts on hate speech detection. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 18, pages 2025–2039.

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.

Zhipeng Zhou, Xingnan Zhou, Yudi Chen, and Haonan Qi. 2024. Evolution of online public opinions on major accidents: Implications for post-accident response based on social media network. *Expert Systems with Applications*, 235:121307.

7 Appendix A

7.1 Advanced Attention Mechanisms

7.1.1 Gated Adaptive Weighting

The gated adaptive weighting component combines the context vectors at the word and sentence level using a gating mechanism.

$$c_{combined} = [c_w; c_s] \tag{15}$$

$$g = \sigma(W_g^2 \tanh(W_g^1 c_{combined} + b_g^1) + b_g^2) \tag{16}$$

$$c_{final} = g \cdot c_w + (1 - g) \cdot c_s \tag{17}$$

where $W_g^1 \in \mathcal{R}^{D\times 2D}$, $W_g^2 \in \mathcal{R}^{1\times D}$, $b_g^1 \in \mathcal{R}^D$, and $b_g^2 \in \mathcal{R}$ are learnable parameters, $g \in \mathcal{R}$ is the gate value, σ is the sigmoid function, and $c_{final} \in \mathcal{R}^D$ is the final context vector that balances word and sentence-level information. This gated adaptive weighting mechanism allows our model to dynamically adjust the importance of word-level and sentence-level features for each input, potentially improving its ability to detect hate speech across various linguistic contexts.

7.1.2 Multi-head Adaptive Weighting

Multi-head methods incorporate adaptive weighting within their structure through multiple heads, without the need for an additional weighting step. After obtaining the context vectors for each head, we concatenate them and apply an adaptive weighting mechanism:

$$c_{combined} = [c_w^1; c_s^1; c_w^2; c_s^2; ...; c_w^h; c_s^h] \tag{18}$$

$$\beta = \text{softmax}(W_a^2 \text{ReLU}(W_a^1 c_{combined} + b_a^1) + b_a^2) \tag{19}$$

$$c_{final} = \sum_{i=1}^{2h} \beta_i c_i \tag{20}$$

where $W_a^1 \in \mathcal{R}^{D\times 2hD}$, $W_a^2 \in \mathcal{R}^{2h\times D}$, $b_a^1 \in \mathcal{R}^D$, and $b_a^2 \in \mathcal{R}^{2h}$ are learnable parameters, $\beta \in \mathcal{R}^{2h}$ are the adaptive weights, and $c_{final} \in \mathcal{R}^D$ is the final context vector.

7.1.3 Multi-head Gated Adaptive Weighting

The Multi-Head Gated Adaptive Weighting (MH-GAW) mechanism extends the concept of adaptive weighting by using multiple attention heads and incorporating a gating mechanism. This approach allows the model to capture different aspects of the input simultaneously and dynamically balance the importance of word-level and sentence-level features. For each head i (where $i = 1, 2, ..., h$, and h is the number of heads):

$$e_w^i = \tanh(W_w^{1i} H + b_w^{1i}) \tag{21}$$

$$\alpha_w^i = \text{softmax}(W_w^{2i} e_w^i + b_w^{2i}) \tag{22}$$

$$c_w^i = \sum_{j=1}^{L} \alpha_w^{ij} H^j \tag{23}$$

$$e_s^i = \tanh(W_s^{1i} H + b_s^{1i}) \tag{24}$$

$$\alpha_s^i = \text{softmax}(W_s^{2i} e_s^i + b_s^{2i}) \tag{25}$$

$$c_s^i = \sum_{j=1}^{L} \alpha_s^{ij} H^j \tag{26}$$

where $W_w^{1i}, W_w^{2i}, W_s^{1i}, W_s^{2i}$ are learnable parameters for each head, $\alpha_w^i, \alpha_s^i \in \mathcal{R}^L$ are the attention weights, and $c_w^i, c_s^i \in \mathcal{R}^D$ are the word-level and sentence-level context vectors for each head. The context vectors from all heads using equation 27 and 28

$$c_w = \frac{1}{h} \sum_{i=1}^{h} c_w^i \tag{27}$$

$$c_s = \frac{1}{h} \sum_{i=1}^{h} c_s^i \tag{28}$$

where $c_w, c_s \in \mathcal{R}^D$ are the aggregated word-level and sentence-level context vectors. A gating mechanism is applied to dynamically balance the word-level and sentence-level information using equations 29, 30 and 31

$$c_{combined} = [c_w; c_s] \tag{29}$$

$$g = \sigma(W_g^2 \tanh(W_g^1 c_{combined} + b_g^1) + b_g^2) \tag{30}$$

$$c_{gated} = g_1 \cdot c_w + g_2 \cdot c_s \tag{31}$$

where $W_g^1 \in \mathcal{R}^{D\times 2D}$, $W_g^2 \in \mathcal{R}^{2\times D}$, $b_g^1 \in \mathcal{R}^D$, and $b_g^2 \in \mathcal{R}^2$ are learnable parameters, $g \in \mathcal{R}^2$ are the gate values, σ is the sigmoid function, and $c_{gated} \in \mathcal{R}^D$ is the gated context vector and final context vector passed in classification head.

8 Appendix B

8.1 Task B Results

The performance in Subtask B using the Hierarchical Adaptive Attention Model is shown in Table 4. We achieved an accuracy of 0.73 on the test set of 4,076 samples. For class 0 (non-hate speech), the model attained precision of 0.93, recall of 0.94, and an F1-score of 0.94 across 3,602 instances. For class 1 (hate speech), it recorded precision of 0.52, recall of 0.46, and an F1-score of 0.49 over 474 instances.

The macro-averaged F1-score was 0.71, and the weighted F1-score was 0.88. This indicates a significant disparity in performance between classes, with high effectiveness in detecting non-hate speech (F1: 0.94) due to the model's handling of the majority class. However, detecting hate speech was more challenging (F1: 0.49), showing that while precision (0.52) and recall (0.46) were balanced, the minority class proved difficult to classify accurately.

Table 4: Results for Subtask B: Hate Speech Detection Using Hierarchical Adaptive Attention Model

	prec.	**rec.**	**f1**	**supp.**
0	0.93	0.94	0.94	3602
1	0.52	0.46	0.49	474
acc.			0.73	4076
macro	0.72	0.70	0.71	4076
weighted	0.88	0.89	0.88	4076

The performance in the Subtask B using the Hierarchical Gated Adaptive Attention Model is presented in Table 5. The model achieved an accuracy of 0.86 on the test set of 4,076 samples. For class 0, the model attained precision of 0.95, recall of 0.90, and an F1-score of 0.92 across 3,602 instances. For class 1, the precision was 0.44, recall was 0.63, and F1-score was 0.52 over 474 instances. The macro-averaged and weighted F1-score was 0.72, and 0.87 respectively.

This architecture demonstrated strong overall performance with 86% accuracy and improved handling of class imbalance. Non-hate speech detection remained high (F1: 0.92), with excellent precision (0.95) and a slightly lower recall (0.90). Also in hate speech detection, we observed an increase of 0.63 in recall, although precision dropped to 0.44, indicating better minority class detection at the cost of some additional false positives. The macro F1-score of 0.72 and weighted F1-score of 0.87 reflect robust performance and enhanced handling of the minority class.

Table 5: Results for Subtask B: Hate Speech Detection Using Hierarchical Gated Adaptive Attention Model

	prec.	**rec.**	**f1**	**supp.**
0	0.95	0.90	0.92	3602
1	0.44	0.63	0.52	474
acc.			0.86	4076
macro	0.70	0.76	0.72	4076
weighted	0.89	0.86	0.87	4076

The performance in Subtask B using the Hierarchical Multi-head Adaptive Weighting model is shown in Table 6. The model achieved an accuracy of 0.89 on a Val set of 4,076 samples. For class 0, the model recorded precision of 0.92,recall of 0.95, and an F1-score of 0.94 across 3,602 instances. For class 1, the precision was 0.52, the recall was 0.39, and the F1-score was 0.45 over 474 instances. The macro-averaged F1-score was 0.69, while the weighted F1-score was 0.88.

While achieving the highest accuracy at 89%, the model displayed significant class imbalance in performance. Non-hate speech detection was highly effective. However, hate speech detection struggled (F1: 0.45), with a low recall (0.39), indicating missed detections despite moderate precision (0.52). The macro F1-score of 0.69 reflects the challenge of achieving balanced performance across classes, while the high weighted F1-score of 0.88 underscores strong performance on the majority class.

Table 6: Results for Subtask B: Hate Speech Detection Using Hierarchical Multi-head Adaptive Weighting

	prec.	**rec.**	**f1**	**supp.**
0	0.92	0.95	0.94	3602
1	0.52	0.39	0.45	474
acc.			0.89	4076
macro	0.72	0.67	0.69	4076
weighted	0.88	0.89	0.88	4076

The performance in Subtask B using the Hierarchical Multi-head Gated Adaptive Weighting model is presented in Table 7. The model achieved an accuracy of 0.87 on a val set of 4,076 samples. For class 0, it demonstrated precision of 0.94, recall of 0.91, and an F1-score of 0.93 across 3,602 instances. For class 1, it recorded precision of 0.45,recall of 0.58, and an F1-score of 0.51 over 474 instances. The macro and weighted averaged F1-score was 0.72 and 0.88 respectively.

The model achieved 87% accuracy with improved balance across classes. Non-hate speech detection remained strong, showing balanced precision and recall. Hate speech detection improved (F1: 0.51) with increased recall (0.58) compared to the non-gated version, though precision was moderate (0.45). The macro F1-score of 0.72 matches that of the gated attention model, indicating comparable balanced performance.

Table 7: Results for Subtask B: Hate Speech Detection Using Hierarchical Multi-head Gated Adaptive Weighting

	prec.	**rec.**	**f1**	**supp.**
0	0.94	0.91	0.93	3602
1	0.45	0.58	0.51	474
acc.			0.87	4076
macro	0.70	0.75	0.72	4076
weighted	0.89	0.87	0.88	4076

In subtask B, we analyzed different proposed architectures to gain crucial insights. The Hierarchical Multi-head Adaptive Weighting model achieved the highest accuracy of 89% but showed weak performance in detecting the minority class. In contrast, the Hierarchical Gated Adaptive Attention model provided a more balanced performance, with 86% accuracy and significantly improved hate speech detection, while maintaining strong non-hate speech detection. Both gated architectures, Gated Adaptive Attention and Multi-head Gated Adaptive Weighting models consistently outperformed their non-gated counterparts in minority class detection, achieving macro F1 scores of 0.72. The substantial class imbalance influenced model behavior, with gating mechanisms proving particularly effective in managing this challenge. These results indicate that, for practical applications in binary hate speech detection, gated architectures offer optimal performance by balancing overall accuracy with reliable minority class detection.

8.2 Task C Results

The performance of the hierarchical adaptive weighting model for identifying hate speech targets (Subtask C) categorized as ’individual,’ ’organization,’ or ’community’—is presented in Table 8. The model achieved accuracy of 73% on val set of 474 samples. For class 0 (individual), it reported precision of 0.78, recall of 0.76, and F1-score of 0.77 across 230 instances. For class 1 (organization), the precision was 0.75, recall was 0.78, and F1-score was 0.76 over 183 instances. For class 2 (community), the model had precision of 0.45, recall of 0.44, and F1-score of 0.45 over 61 instances. The macro-averaged F1-score was 0.66, and the weighted F1-score matched the accuracy at 0.73.

While achieving balanced macro-averaged precision and recall of 0.66, the model performed well for individual (F1: 0.77) and organizational targets (F1: 0.76). However, identifying community targets remained challenging, with both precision and recall at 0.45, indicating difficulty in detecting the minority class. The weighted F1-score of 0.73 reflects the model’s proportionate performance across the class distributions. The confusion matrix, shown in Figure 2, provides deeper insight into the model performance across the classes.

The Gated Adaptive Weighting model, in Subtask C achieved an accuracy of 75% on a test set of 474 samples, as shown in Table 9. For class 0, the model reported precision of 0.78, recall of 0.81, and F1-score of 0.79 across 230 instances. For

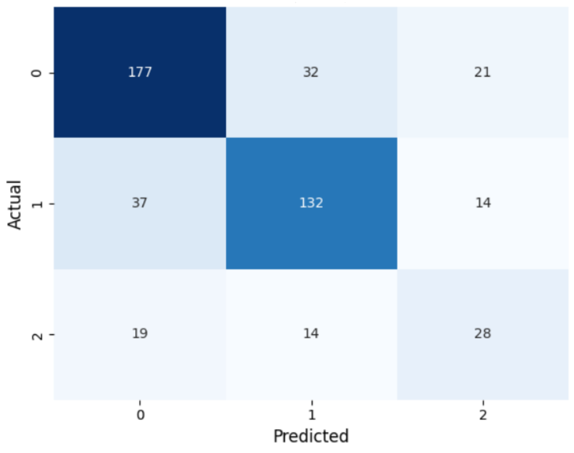


Figure 2: Confusion matrix of the hierarchical adaptive weighting model.

Table 8: Results for Subtask C: Hate Speech Detection Using Hierarchical Adaptive Weighting

	Prec.	Rec.	F1	Supp.
0	0.78	0.76	0.77	230
1	0.75	0.78	0.76	183
2	0.45	0.44	0.45	61
Acc.			0.73	474
Macro	0.66	0.66	0.66	474
Weight.	0.73	0.73	0.73	474

class 1, it recorded precision of 0.80, recall of 0.75, and F1-score of 0.77 across 183 instances. For class 2 , the model attained precision of 0.51,recall of 0.52, and F1-score of 0.52 across 61 instances. The macro-averaged F1-score was 0.69, while the weighted F1-score was 0.75.

The gating mechanism significantly enhanced minority class detection, resulting in an F1-score of 0.52 with balanced precision and recall of 0.51 and 0.52, respectively. Performance on individual targets improved, achieving an F1-score of 0.79 (precision: 0.78, recall: 0.81), while organizational target detection reached an F1-score of 0.77 (precision: 0.80, recall: 0.75). The model demonstrated a balanced precision-recall trade-off across all classes, with weighted metrics consistently at 0.75, indicating robust performance regardless of class distribution. Figure 3 presents the confusion matrix, offering a detailed view of the performance of the model in all classes.

The Multi-head Adaptive Weighting model in Subtask C achieved an accuracy of 73% on val set of 474 samples, as shown in Table 10. For class 0, the model reported precision of 0.78, recall of 0.78, and F1-score of 0.78 across 230 instances. For class 1, it recorded precision of 0.76, recall of

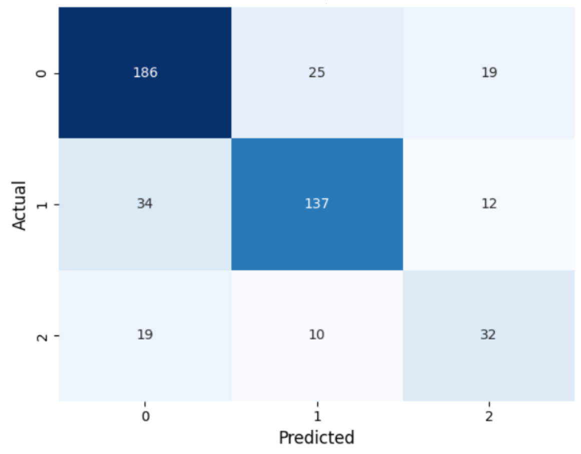


Figure 3: Confusion matrix of the hierarchical gated adaptive weighting model.

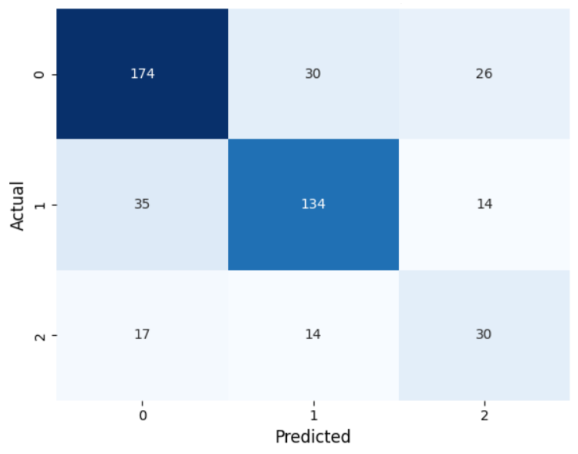


Figure 4: Confusion matrix of the multi-head adaptive weighting model.

Table 9: Results for Subtask C: Hate Speech Detection Using Hierarchical Gated Adaptive Weighting

	Prec.	Rec.	F1	Supp.
0	0.78	0.81	0.79	230
1	0.80	0.75	0.77	183
2	0.51	0.52	0.52	61
Acc.			0.75	474
Macro	0.69	0.69	0.69	474
Weight.	0.75	0.75	0.75	474

Table 10: Results for Subtask C: Hate Speech Detection Using Muti-head Adaptive Weighting

	Prec.	Rec.	F1	Supp.
0	0.78	0.78	0.78	230
1	0.76	0.73	0.74	183
2	0.47	0.52	0.50	61
Acc.			0.73	474
Macro	0.67	0.68	0.67	474
Weight.	0.73	0.73	0.73	474

0.73, and F1-score of 0.74 across 183 instances. For class 2 , the model attained a precision of 0.47, a recall of 0.52, and an F1-score of 0.50 across 61 instances. The macro and weighted F1-scores were 0.67 and 0.73, respectively.

With an accuracy of 73%, this model demonstrated a slight improvement in recall (macro-recall: 0.68). Detection of individual targets maintained strong performance (F1: 0.78), with both precision and recall at 0.78. For organizational targets, there was a slight decline (F1: 0.74), with precision at 0.76 and recall at 0.73. Community target detection improved moderately compared to the non-gated hierarchical model, achieving an F1-score of 0.50, with precision at 0.47 and recall at 0.52. The weighted metrics stabilized at 0.73, consistent with accuracy. The confusion matrix depicted in Figure 4 sheds light on the model's performance for each class.

The Multi-head Gated Adaptive Weighting model in Subtask C achieved an accuracy of 73% on a val set of 474 samples, as shown in Table 11. For class 0, the model achieved precision of 0.76,recall of 0.78, and an F1-score of 0.77 across 230 instances. For class 1, it recorded a precision of 0.78, a recall of 0.75, and an F1-score of 0.76 across 183 instances. For class 2, the model attained precision of 0.43,recall of 0.43, and F1-score of 0.43 across 61 instances. The macro-averaged F1-score was 0.66, while the weighted F1-score was 0.73.

Despite its architectural sophistication, the model performance remained at 73% accuracy. It showed a regression in minority class detection (F1: 0.43), with balanced but lower precision and recall (both at 0.43). Individual target detection maintained effectiveness (F1: 0.77) with a precision of 0.76 and a recall of 0.78. Organizational targets exhibited similar results (F1: 0.76) with a precision of 0.78 and a recall of 0.75. Detailed information on the model's classification accuracy across the classes can be observed in the confusion matrix shown in Figure 5)

Table 11: Results for Subtask C: Hate Speech Detection Using Muti-head Gated Adaptive Weighting

	Prec.	Rec.	F1	Supp.
0	0.76	0.78	0.77	230
1	0.78	0.75	0.76	183
2	0.43	0.43	0.43	61
Acc.			0.73	474
Macro	0.66	0.65	0.66	474
Weight.	0.73	0.73	0.73	474

Comparative analysis identifies the Gated Adap-

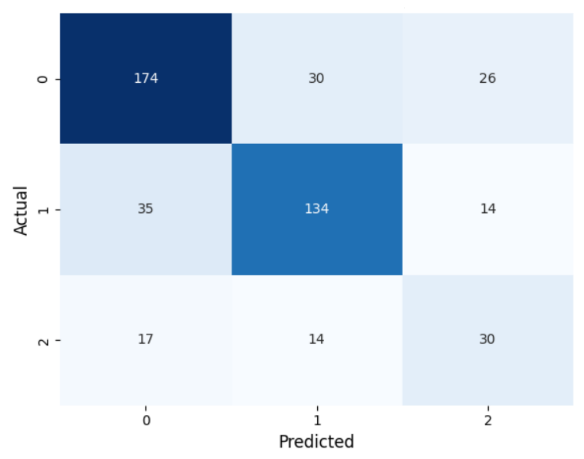


Figure 5: Confusion matrix of the multi-head gated adaptive weighting model.

tive Weighting model as the optimal architecture, demonstrating superior performance across all evaluation metrics. The introduction of gating mechanisms significantly enhanced minority class detection while maintaining strong performance on majority classes. In contrast, the multi-head approaches, despite their complexity, did not yield substantial improvements over simpler architectures. A persistent challenge across all models is the detection of community-targeted hate speech, indicating need for additional techniques to address class imbalance. These results suggest that architectural simplicity, combined with effective feature selection through gating, outperforms more complex attention mechanisms in hate speech and target detection.

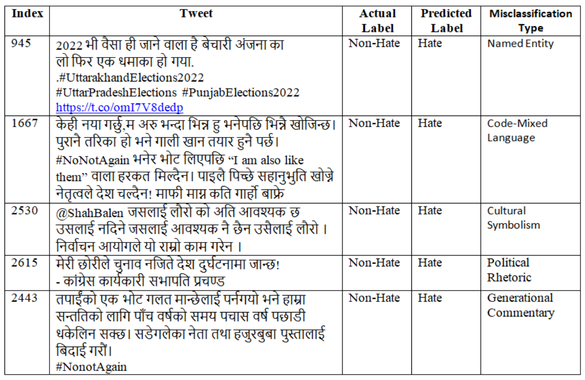
9 Appendix C (Error Analysis)

9.1 Appendix C -I

The model exhibited specific challenges in classifying tweets containing symbolic language, named entities, and code-mixed expressions, particularly in understanding nuanced cultural and script-specific references in Devanagari-based low-resource languages. Figure 6 shows representative examples that illustrate these limitations:

The tweet(Index 945) references “Anjana” in a non-hateful, casual context within election commentary. However, the model incorrectly flags this as hate speech, demonstrating difficulties in interpreting named individuals in colloquial, non-aggressive contexts.

The tweet(Index 1667) critiques leadership, which could be interpreted as negative, the language does not explicitly target any individual or group with hate speech.The hashtag "#NoNotAgain" is often associated with resistance or opposition, which may have contributed to the model’s identification of the tweet as possibly related to hate speech, especially in the context of political discourse. This tweet contains a mix of Nepali and English, phrase as a political expression of discontent without hate. The model’s misclassification reveals limitations in processing symbolic expressions within code-mixed language, especially involving Devanagari script. The tweet(Index 2530) contains the term ‘"lauro" carries symbolic meaning in Nepali political discourse. Used here in a non-hostile critique of electoral commission decisions, its misclassification as hate speech highlights the model’s difficulty in interpreting culturally specific metaphors without explicit hate markers. The tweet("Index 2615") contains hyperbolic political rhetoric attributed to a political figure. The model’s incorrect classification demonstrates challenges in processing sarcasm and layered interpretation. The language used in the tweet(Index 2443), such as "Galat manchhelai parnayo" (voting for the wrong person) and "sadegleka neta" (rotten leaders), carries a negative tone. This could be seen as aggressive or disrespectful towards political leaders, which may have influenced the model’s classification.However, while the tone is critical, the tweet doesn’t contain hate speech towards any particular group. The phrase "Hajurbuba pustalai bidayi garaun" (let’s say goodbye to the older generation of leaders) focuses on generational change. The critique is directed at political figures considered outdated or ineffective, which is a common political sentiment.The model could have misinterpreted the critical nature of the tweet as hateful, possibly because of the phrases "Galat manchhelai parnayo",



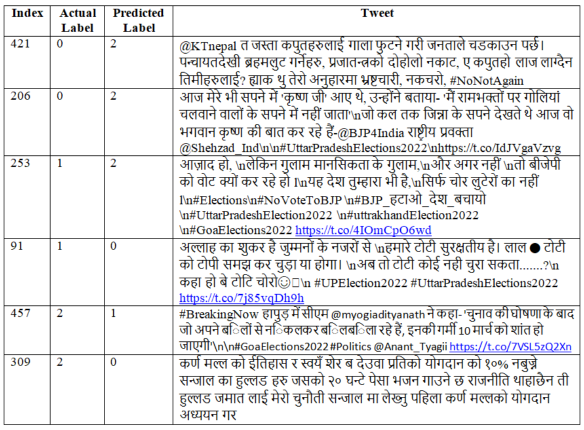
Index	Tweet	Actual Label	Predicted Label	Misclassification Type
945	2022 भी वैसा ही जाने वाला है बेचारी अंजना का लो फिर एक धमाका हो गया. .#UttarakhandElections2022 #UttarPradeshElections #PunjabElections2022 https://t.co/omI7V8dedp	Non-Hate	Hate	Named Entity
1667	केही नया गर्छु,म अरु भन्दा भिन्न हु भनेपछि भिन्नै खोजिन्छ। पुरानै तरिका हो भने गाली खान तयार हुनै पर्छ। #NoNotAgain भनेर भोट लिएपछि “I am also like them” वाला हरकत मिल्दैन। पाइलै पिच्छे सहानुभुति खोज्ने नेतृत्वले देश चल्दैन! माफी माग्न कति गार्हो बाफ्रे	Non-Hate	Hate	Code-Mixed Language
2530	@ShahBalen जसलाई लौरो को अति आवश्यक छ उसलाई नदिने जसलाई आवश्यक नै छैन उसैलाई लौरो । निर्वाचन आयोगले यो राम्रो काम गरेन ।	Non-Hate	Hate	Cultural Symbolism
2615	मेरी छोरीले चुनाव नजिते देश दुर्घटनामा जान्छ! - कांग्रेस कार्यकारी सभापति प्रचण्ड	Non-Hate	Hate	Political Rhetoric
2443	तपाईंको एक भोट गलत मान्छेलाई पर्नगयो भने हाम्रा सन्ततिको लागि पाँच वर्षको समय पचास वर्ष पछाडी धकेलिन सक्छ। सडेगलेका नेता तथा हजुरबुबा पुस्तालाई बिदाई गरौं। #NonotAgain	Non-Hate	Hate	Generational Commentary

Figure 6: Error analysis of samples of Substask B

"Hajurbuba pustalai bidayi garaun" and "sadegleka neta" (rotten leaders).

9.2 Appendix C -II

To evaluate our model's robustness in distinguishing hate speech targets, we conducted an error analysis on misclassified instances. This analysis provided insights into common misclassification patterns, particularly among categories of individuals, organizations, and communities. Figure 7 presents examples of these errors, along with interpretations for each case. The analysis revealed specific trends in misclassification across categories. For instance, tweets targeting individuals (Class 0) were frequently misclassified as targeting communities (Class 2), likely due to language that generalized statements to a broader group. Similarly, tweets aimed at organizations (Class 1) were often misinterpreted as targeting communities (Class 2) due to the use of collective or broad descriptors.



Index	Actual Label	Predicted Label	Tweet
421	0	2	@KTnepal त जस्ता कपुतहरुलाई गाला फुटने गरी जनताले चडकाउन पर्छ। पन्चायतदेखी ब्रहमलुट गर्नेहरु, प्रजातन्त्रको दोहोलो नकाट, ए कपुतहो लाज लाग्दैन तिमीहरुलाई? ह्याक थु तेरो अनुहारमा भ्रष्टचारी, नकचरो, #NoNotAgain
206	0	2	आज मेरे भी सपने में 'कृष्ण जी' आए थे, उन्होंने बताया- 'मैं रामभक्तों पर गोलियां चलवाने वालों के सपने में नहीं जाता'\nजो कल तक जिन्ना के सपने देखते थे आज वो भगवान कृष्ण की बात कर रहे हैं-@BJP4India राष्ट्रीय प्रवक्ता @Shehzad_Ind\n\n#UttarPradeshElections2022\nhttps://t.co/IdJVgaVzvg
253	1	2	आज़ाद हो, \nलेकिन गुलाम मानसिकता के गुलाम,\nऔर अगर नहीं \nतो बीजेपी को वोट क्यों कर रहे हो l\nयह देश तुम्हारा भी है,\nसिर्फ चोर लुटेरों का नहीं l\n#Elections\n#NoVoteToBJP \n#BJP_हटाओ_देश_बचायो \n#UttarPradeshElection2022 \n#uttrakhandElection2022 \n#GoaElections2022 https://t.co/4IOmCpO6wd
91	1	0	अल्लाह का शुकर है जुम्मनों के नजरों से \nहमारे टोटी सुरक्षतीय है। लाल ● टोटी को टोपी समझ कर चुड़ा या होगा। \nअब तो टोटी कोई नही चुरा सकता.......?\n कहा हो बे टोटि चोरो☺□\n #UPElection2022 #UttarPradeshElections2022 https://t.co/7j85vqDh9h
457	2	1	#BreakingNow हापुड़ में सीएम @myogiadityanath ने कहा- 'चुनाव की घोषणा के बाद जो अपने ब िलों से न िकलकर ब िलब िला रहे हैं, इनकी गर्मी 10 मार्च को शांत हो जाएगी'\n\n#GoaElections2022 #Politics @Anant_Tyagii https://t.co/7VSL5zQ2Xn
309	2	0	कर्ण मल्ल को ईतिहास र स्वयँ शेर ब देउवा प्रतिको योगदान को १०% नबुझ्ने सन्जाल का हुल्लड हरु जसको २० घन्टे पेसा भजन गाउने छ राजनीति थाहाछैन ती हुल्लड जमात लाई मेरो चुनौती सन्जाल मा लेख्नु पहिला कर्ण मल्लको योगदान अध्ययन गर

Figure 7: Error analysis of samples of Substask C

In tweet (Index 421), a specific individual "@KTnepal" is criticized using direct language, but the inclusion of broader terms like "Prajatantr ko dohol na kaat" (loosely criticizing broader democratic practices) likely led the model to misclassify it as targeting a community (Class 2). Tweet (Index 206) targets a spokesperson (an individual) but also references religious and political groups ("Rambhakton" and "Jinnah"), which may have confused the model into categorizing it as community-level speech (Class 2). Tweet (Index 253) critiques the BJP (a political organization) but uses phrases like "Gulaam mansikta ke gulaam" ("slave mentality"), which could be interpreted as a critique of a broader societal mindset. This likely led the model to classify it under communities (Class 2) instead of organizations (Class 1). Tweet (Index 91) employs sarcastic commentary that seems directed at an individual due to its personal tone ("Jummon ke nazron se hamari totee surakshit hai"), but it actually targets a group associated with a particular ideology. The model misinterpreted this, resulting in a classification as an individual (Class 0) rather than an organization (Class 1).

In tweet (Index 457), although the statement references a community ("Jo apne bilon se nikal kar bilbila rahe hain"), it is attributed to a political leader (@myogiadityanath). This association with an organization might have caused the model to misclassify it as targeting an organization (Class 1) instead of a community (Class 2). The tweet (Index 309) criticizes a group but also mentions a specific individual, "Karn Mall." The presence of this individual reference may have confused the model, leading to a classification under individuals (Class 0) rather than communities (Class 2). These findings suggest that our model requires improved contextual awareness, particularly in handling nuanced linguistic features like collective nouns and generalized rhetoric. Future iterations could benefit from incorporating additional context markers or keywords associated with specific entities to enhance target classification accuracy."