Gendered Abuse Detection in Indic Languages

Satyam Singh[1], Sankaranarayanan Sengunther[1], and Kushal Mitra[1]

[1]IIIT Delhi , {satyam24082, sankaranarayanan24081, kushal24050}@iiitd.ac.in

1 Introduction

Online spaces, much like their offline counterparts, are not immune to the impact of gender-based harassment. In fact, the digital realm has become a fertile ground for the continued expression of sexism, often in the form of abusive and exclusionary language. This persistent issue significantly hampers individuals' freedom of expression, reinforcing a culture of inequality and psychological harm. As explored by Whiley et al. (2023) and Hoskin and Whiley (2023), the online propagation of gendered slurs and toxic dialogue plays a central role in fostering harassment and shaping discriminatory narratives. Importantly, these dynamics do not only affect women—men, too, are impacted by the pressures of toxic masculinity, which leaves behind emotional scars and undermines mental well-being. Research by Kural and Kovács (2022) highlights the links between online abuse and mental health challenges such as anxiety, low self-worth, and lingering insecurity. Similarly, Feigt et al. (2022) point out the long-term toll such abuse can take on both personal relationships and psychological stability, while Barreto and Doyle (2023) underscore how the repetition of gender-biased language reinforces systemic inequality.

In light of these challenges, the development of automated tools to detect and reduce sexist and abusive language is not just beneficial—it is necessary. These systems hold the potential to curb the reach of harmful content and promote safer, more inclusive digital communities (Vetagiri et al.). Yet, identifying gendered abuse online remains a complex task. Language is deeply contextual, often informal, and frequently interwoven with cultural nuance and code-switching—especially on platforms like Twitter (Van Dijk, 2015). This complexity becomes even more pronounced in Indic language contexts, where limited digital resources, hybrid expressions, and language mixing present unique obstacles to automated moderation.

To address this gap, we introduce a system for Gendered Abuse Detection in Indic Languages, designed to identify sexist and abusive content in English, Hindi, and Tamil. Our work builds on a diverse dataset of over 18,000 labeled tweets—each language contributing more than 6,000 examples—including a newly annotated collection curated by Arora et al. (2023) for the ICON 2023 shared task, organized by Tattled Civic Tech. This dataset includes 6532 English, 6198 Hindi, and 6780 Tamil tweets, annotated by a team of 18 researchers and activists who based their insights on authentic instances of online gendered harassment.

To tackle the detection task, we implement a combination of deep learning strategies capable of capturing both linguistic subtleties and contextual patterns. Our architecture leverages BERT-based embeddings and transformer models, incorporating a GRU enhanced with restricted self-attention for fine-grained feature extraction, alongside a Transformer with Gated CNNs to enhance contextual understanding. These choices are particularly effective for parsing the noisy, multilingual nature of social media posts. When benchmarked against leading models like mBERT and IndicBERT, our system demonstrates consistent performance gains. In particular, it outperforms the CNLP-NITS-PP model and a custom transformer with restricted self-attention across all evaluation tasks, setting a new standard for gendered abuse detection in Indic language settings.

2 Related Work

In recent years, there has been growing interest in developing automated systems to detect sexist and abusive content online—a reflection of its serious implications for individual mental health and the broader safety of digital communities. Earlier approaches in this area primarily depended on rule-based methods and conventional machine learning models such as Support Vector Machines (SVM) and Naive Bayes. These techniques often utilized handcrafted linguistic features, including lexical patterns and syntactic cues. While effective in limited settings, they frequently fell short in generalizing to informal, multilingual platforms like social media, where users commonly engage in code-switching and use highly varied, non-standard language (Waseem and Hovy, 2016; Nobata et al., 2016).

To overcome these limitations, researchers began turning to deep learning models—particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). These architectures offered better capabilities in learning from noisy, unstructured textual data and delivered stronger performance when trained on sufficiently large, labeled corpora. Nonetheless, these models still encountered difficulties when attempting to identify more subtle or contextually embedded abusive language (Zhou et al., 2016; Pavlopoulos et al., 2017).

A major breakthrough came with the advent of transformer-based models like BERT (Devlin et al., 2019). These models leveraged pretrained contextual embeddings, enabling them to capture word meanings more accurately within varied grammatical contexts. The emergence of multilingual transformers such as mBERT and XLM-R has further pushed the boundaries, showing promising results in detecting hate speech and abusive language across diverse languages—including those with limited resources (Röttger et al., 2021). Specifically for Indian languages, IndicBERT (Kakwani et al., 2020) has demonstrated significant effectiveness across multiple natural language processing tasks, including detecting online abuse.

Despite these advancements, key challenges persist when working with Indic languages. These include a scarcity of annotated data, the prevalence of code-mixed communication, and the cultural nuance often embedded in gendered insults. In our approach, we propose an architecture that combines the strengths of two specialized deep learning models to effectively capture both local and global linguistic features. The first component is a GRU-based model equipped with restricted self-attention, designed to focus on short-span patterns indicative of abuse. The second component is a Transformer enhanced with Gated CNNs, allowing it to grasp broader contextual relationships as well as hierarchical n-gram structures. Both models are further powered by contextual embeddings from pretrained language models—mBERT and IndicBERT—enabling nuanced semantic understanding across the three languages under study: English, Hindi, and Tamil.

3 Methodology

3.1 Baseline Models

To set a solid foundation for evaluating our proposed approach, we developed a series of baseline models covering all three tasks in our study:

- **Task 1:** Classification of gendered abuse
- **Task 2:** Transfer learning for hate speech detection
- **Task 3:** Multi-task learning to identify both gendered abuse and explicit content

These baselines include both established pretrained models and custom-designed neural network architectures.

3.1.1 Pretrained Transformer Models

Task 1 – mBERT Fine-Tuning We began by fine-tuning a multilingual BERT (mBERT) model to detect gendered abuse in a combined dataset comprising tweets in English, Hindi, and Tamil. The dataset was created by merging the three language-specific corpora, applying preprocessing techniques, and assigning labels using a majority-vote approach. This baseline achieved a Macro F1-score of 0.7940, and the final model was saved as a Hugging Face-compatible checkpoint.

Task 2 – Transfer Learning with Pretrained mBERT For the second task, we adopted a transfer learning strategy using an mBERT model that had already been trained on hate speech detection. We further fine-tuned it on the same multilingual dataset used in Task 1. The preprocessing steps remained consistent, including text normalization and majority voting. This model attained a Macro F1-score of 0.656 and was also exported as a Hugging Face-compatible checkpoint.

Task 3 – IndicBERT for Multi-Task Learning To simultaneously detect gendered abuse and explicit language, we built a multi-task model using IndicBERT. The model utilized a shared encoder to extract common features, while separate fully connected layers handled each task independently. After preprocessing and applying the majority vote for label consistency, this model reached a Macro F1-score of 0.76. The trained checkpoint is also available for downstream use.

3.1.2 Custom Neural Baselines

Tasks 1 & 2 – Transformer with Restricted Self-Attention We also explored a custom transformer architecture equipped with restricted self-attention, aimed at capturing short-range abusive patterns often found in social media text. This model recorded a Macro F1-score of 69.90% on Task 1 with 431 false negatives, and 60.94% on Task 2 with 467 false negatives.

Task 3 – GRU with Restricted Self-Attention For the third task, we implemented a GRU-based architecture enhanced with restricted self-attention. This model achieved an average Macro F1-score of 0.668. Breaking down performance per subtask:

- For Question 1, the Macro F1-score was 0.6008, with 331 false negatives
- For Question 3, it achieved 0.7328, with 586 false negatives

3.1.3 Reproduction of Prior Work

To further benchmark our results, we reproduced a model inspired by prior research combining Convolutional Neural Networks (CNNs) and Bidirectional LSTMs (BiLSTM), using GloVe embeddings. The CNN layer captured local features, while the BiLSTM layer helped model sequential dependencies. The model was trained on a dataset of over 6,000 English-language tweets and optimized using stratified K-Fold cross-validation, early stopping, and dynamic learning rate adjustment. On Task 1, this model delivered a Precision of 0.81, Recall of 0.76, and an F1-score of 0.77 on the validation set.

3.2 Proposed Architecture

3.2.1 GRU with Restricted Self-attention

To balance modeling power with computational efficiency, we adopt a hybrid architecture that combines frozen multilingual BERT embeddings with bidirectional GRUs and a custom windowed self-attention mechanism. The model begins by using BERT to generate rich contextual representations of input tokens, without fine-tuning, which reduces training overhead. These embeddings are then passed through stacked GRU layers that process the sequence bidirectionally, capturing temporal dependencies in both directions.

What sets this model apart is its windowed self-attention layer, which allows each token to attend only to a fixed local neighborhood within the sequence. This targeted attention mechanism captures nearby dependencies more efficiently than global self-attention, while still enabling dynamic context aggregation. After attention, we apply mean pooling followed by two dense layers with ReLU activation, dropout, and a final sigmoid function for classification.

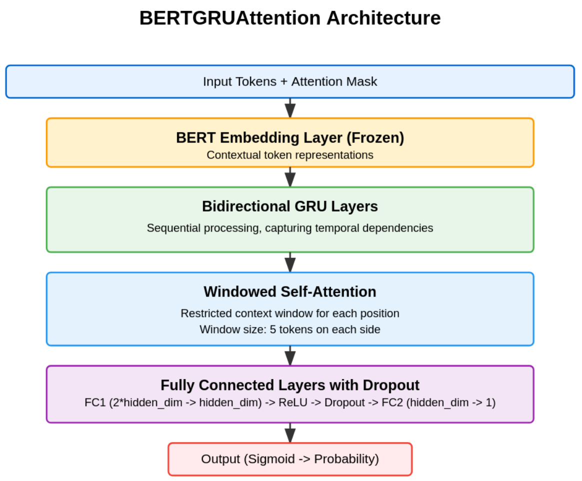
BERTGRUAttention Architecture
Input Tokens + Attention Mask
BERT Embedding Layer (Frozen)
Contextual token representations
Bidirectional GRU Layers
Sequential processing, capturing temporal dependencies
Windowed Self-Attention
Restricted context window for each position
Window size: 5 tokens on each side
Fully Connected Layers with Dropout
FC1 (2*hidden_dim -> hidden_dim) -> ReLU -> Dropout -> FC2 (hidden_dim -> 1)
Output (Sigmoid -> Probability)

Figure 1: Architecture of GRU with restricted Self-attention

Compared to traditional CNN-BiLSTM architectures, this model leverages BERT's stronger pretraining and uses GRUs, which are lighter and faster than LSTMs. In contrast to Transformer-based models with restricted attention, this architecture preserves a sequential inductive bias through GRUs, offering better performance on text tasks that benefit from order-aware modeling. Overall, the combination of pretrained embeddings, efficient sequential processing, and localized attention makes this approach well-suited for detecting nuanced patterns in abusive and gendered content within multilingual social media text.

3.2.2 Transformer with Gate Convolutional Network

Our GCNTransformer architecture blends frozen multilingual BERT embeddings with gated convolutional layers and a transformer encoder to effectively capture both local and global linguistic features. BERT provides rich, contextual token representations without the computational overhead of fine-tuning. These embeddings are passed through a gated convolutional network, which selectively filters information using parallel convolutional layers—one extracting features, the other generating gating values that modulate those features. This mechanism, inspired by Gated Linear Units (GLUs), allows the model to suppress irrelevant signals and emphasize meaningful patterns.

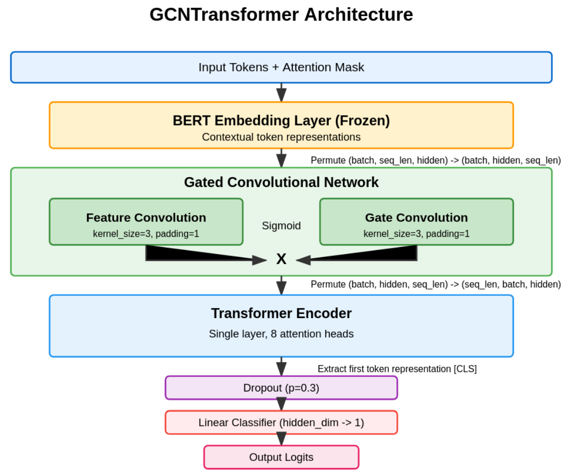
GCNTransformer Architecture
Input Tokens + Attention Mask
BERT Embedding Layer (Frozen)
Contextual token representations
Permute (batch, seq_len, hidden) -> (batch, hidden, seq_len)
Gated Convolutional Network
Feature Convolution
kernel_size=3, padding=1
Sigmoid
Gate Convolution
kernel_size=3, padding=1
X
Permute (batch, hidden, seq_len) -> (seq_len, batch, hidden)
Transformer Encoder
Single layer, 8 attention heads
Extract first token representation [CLS]
Dropout (p=0.3)
Linear Classifier (hidden_dim -> 1)
Output Logits

Figure 2: Architecture of Transformer with GCN

Following this, a single-layer transformer encoder enhances the representations by modeling long-range dependencies across the sequence, enabling each token to attend to others regardless of position. For classification, we use either the [CLS] token or a pooled representation of the sequence, followed by dropout and a final linear layer to produce output logits.

This architecture differs from CNN-BiLSTM setups by replacing sequential modeling with attention-based processing and adding a gating layer for better feature control. Compared to Transformer-based models with restricted attention, the GCNTransformer achieves better local pattern detection through its convolutional front-end while still benefiting from the global attention of the transformer. The result is a computationally efficient yet expressive model, well-suited to detecting nuanced patterns in abusive language.

4 Dataset

For this study, we use a combination of publicly available datasets to train and evaluate our models for detecting gendered abuse in Indic languages. The primary dataset comes from the ICON 2023 Shared Task on Online Gendered Abuse Detection, organized by Tattle Civic Tech. It contains tweets in Indian English, Hindi, and Tamil, annotated by individuals from marginalized gender and LGBTQIA+ communities in South Asia. These annotations are grounded in the annotators' lived experiences and reflect nuanced understandings of abuse in digital spaces.

The dataset includes 6532 posts in Indian English, 6198 in Hindi, and 6780 in Tamil. Each post is labeled across three axes: general gendered abuse, abuse targeting marginalized groups, and whether the content is explicit or aggressive. Annotations may include "1" for yes, "0" for no, "NL" for not labeled, and "NaN" if not assigned. We follow the official train-test splits provided by the shared task for consistency in evaluation.

To support transfer learning and improve the generalization ability of our models, we incorporate additional resources. The MACD dataset offers abusive and non-abusive samples in five Indian languages, while an English-language dataset focused on hate speech and offensive content provides broader linguistic and contextual diversity. These auxiliary datasets help expose our models to varied forms of online abuse, aiding in the development of systems that are both culturally aware and linguistically adaptable.

5 Experimental Setup

The dataset used in this study comprises three distinct labels, each represented across three languages: English, Hindi, and Tamil. For every language and label combination, six human annotators were assigned the task of determining whether a given text instance belonged to the specified label. To ensure annotation reliability, the final label for each instance was determined based on the majority vote among the annotators.

5.1 Preprocessing

Prior to feeding the data into the models, a thorough preprocessing pipeline was applied. All text was converted to lowercase to maintain consistency across samples. Unnecessary elements such as URLs and HTML tags were removed. Additionally, we performed language-specific cleaning, which included the elimination of non-alphabetic characters and extraneous whitespace, ensuring the text remained as linguistically clean and semantically rich as possible.

5.2 Task 1: Gendered Abuse Detection (General)

The objective of the first task was to determine whether a given post contained gendered abuse, even when such abuse was not directly targeted at individuals of marginalized gender identities or sexualities. For this task, all posts labeled as category 1 were collected across English, Hindi, and Tamil, and compiled into a unified dataset.

To address this classification problem, we designed a custom architecture combining a Gated Convolutional Network (GCN) with a Transformer-based encoder. Initially, the text data was tokenized using multilingual BERT (mBERT) embeddings. These embeddings were passed through the GCN, where two types of convolutional operations—standard feature convolutions and gated convolutions—were used to extract contextual features. The resulting feature representations were then forwarded to a Transformer encoder, which provided an enhanced contextual understanding of the input sequence. Finally, the encoded representations were passed to a classifier layer to produce a binary output, indicating whether the text was abusive (1) or not (0).

The model was trained for 10 epochs using the Adam optimizer with a learning rate of 1×10^{-5}, a weight decay of 1×10^{-9}, and a batch size of 16.

5.3 Task 2: Transfer Learning for Abusive Content Detection

The second task focused on leveraging transfer learning to improve the detection of abusive content. In this setup, the model was initially pretrained on external hate speech datasets and subsequently fine-tuned on the ULI dataset to predict whether a given text corresponded to label 1.

For the pretraining phase, different datasets were used depending on the language. The MACD dataset, which includes annotated hate speech data in Hindi and Tamil, served as the pretraining corpus for those two languages. Each instance in this dataset was labeled as either abusive (0) or non-abusive (1). For English, we used the *Hate Speech and Offensive Language Dataset* from Kaggle, where text is categorized into hate speech, offensive language, and neutral content.

The model architecture for this task involved BERT embeddings as the initial input layer. These embeddings were passed through Bidirectional GRU layers to capture both forward and backward contextual dependencies. A restricted self-attention mechanism was applied to enhance the focus on crucial parts of the text. The final attention-weighted outputs were passed through a fully connected layer to predict whether the post was abusive or not.

Training was conducted over 10 epochs with a batch size of 16, using the Adam optimizer and a learning rate of 3×10^{-5}.

5.4 Task 3: Multi-label Detection of Abuse Categories

The final task extended the challenge by requiring the model to detect whether a post lies in label 1 or not and also whether it lies in label 3 or not. For this, data labeled under both categories from all three languages was aggregated into a single CSV file, which was then used for training and evaluation.

The architecture employed for this task was identical to the one used in Task 1, combining Gated Convolutional Networks with a Transformer encoder. The model was trained on the merged dataset, using the same preprocessing techniques described earlier.

Training for this task was carried out over 20 epochs using the AdamW optimizer, with a learning rate of 2×10^{-6}, a weight decay parameter of 0.01, and a batch size of 16.

6 Results

6.1 Evaluation

To evaluate our models' overall performance and efficiency, we adopted several metrics, particularly precision, recall, and the F1 score. The F1 score is an essential evaluation measure because it consolidates the performance of a classifying model in terms of all categories into one statistic by providing a balanced measure between the model's precision and recall.

Precision measures how accurately the model predicts positive outcomes, while recall tells us how well the model captures all the relevant instances, giving us insights into how broadly the models can predict. These metrics are immensely useful in tasks such as classification, where finding a middle ground between accuracy and completeness is crucial.

The precision and recall for the positive class can be calculated as follows:

$$\text{Precision}_1 = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall}_1 = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

Similarly, the precision and recall for the negative class are:

$$\text{Precision}_0 = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{3}$$

$$\text{Recall}_0 = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4}$$

For multi-class classification, the precision and recall for class c are defined as:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \tag{5}$$

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \tag{6}$$

The Macro-Average Precision (MAP) and Macro-Average Recall (MAR) are computed as follows:

$$\text{MAP}_{\text{Binary}} = \frac{\text{Precision}_1 + \text{Precision}_0}{2} \tag{7}$$

$$\text{MAR}_{\text{Binary}} = \frac{\text{Recall}_1 + \text{Recall}_0}{2} \tag{8}$$

$$\text{MAP}_{\text{Multiclass}} = \frac{1}{C}\sum_{c=1}^{C} \text{Precision}_c \tag{9}$$

$$\text{MAR}_{\text{Multiclass}} = \frac{1}{C}\sum_{c=1}^{C} \text{Recall}_c \tag{10}$$

Finally, the Macro F1 score is calculated using:

$$\text{F1}_{\text{macro}} = \frac{2 \times (\text{MAP} \times \text{MAR})}{\text{MAP} + \text{MAR}} \tag{11}$$

To measure the efficiency of our models, we have used the above metrics, the results of which are discussed in the next section.

6.2 Training Results

The training phase results for all tasks were evaluated using the macro F1 score on the validation set. A summary is presented in Table 1, and the loss graphs illustrating model convergence over epochs are shown in Figures 3, 4, and 5.

Table 1: Validation Macro F1 Scores for Training Tasks

Task	Validation Macro F1 Score
Task 1	0.7401
Task 2	0.7100
Task 3 (Q1)	0.6450
Task 3 (Q3)	0.7680

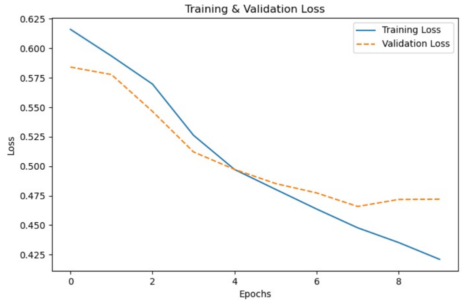

Figure 3: Task 1 Validation Loss

6.3 Testing Results

The testing phase results are presented in Table 2. These include the macro F1 scores for each task, along with confusion matrices visualized in Figures 6, 7, 8, and 9. Notably, the FN (False Negative) counts reveal the number of abusive texts that were incorrectly classified as non-abusive.

Table 2: Testing Macro F1 Scores

Task	Test Macro F1 Score
Task 1	0.7516
Task 2	0.7068
Task 3 (Average)	0.7194
Task 3 (Q1)	0.6585
Task 3 (Q3)	0.7803

6.4 Result Analysis

The outcome of our experiments indicates encouraging performance across the classification tasks outlined in the ICON 2023 shared task. A detailed analysis of the models' behavior during training and testing phases highlights key trends and comparative strengths of different architectures.

In the case of Task 1, the model utilizing a Transformer encoder combined with a Gated Convolutional Network (GCN) demonstrated superior performance. This architecture outperformed both the custom Transformer and the GRU-based models, each integrated with restricted self-attention mechanisms. The Transformer-GCN model achieved a

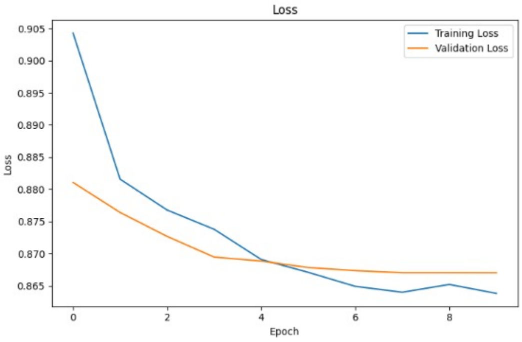

Figure 4: Task 2 Validation Loss

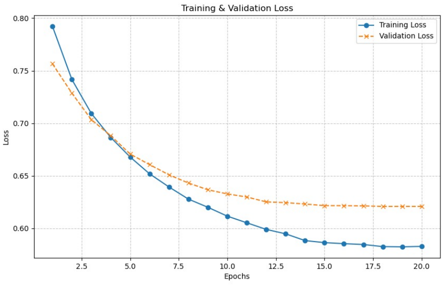

Figure 5: Task 3 Validation Loss

macro F1 score of 0.75, which was notably higher than the 0.699 attained by the custom Transformer and slightly better than the GRU-based model, which scored 0.7403. This suggests that the integration of GCN allowed for more effective feature extraction from multilingual textual data, enhancing the model's discriminatory ability in identifying abusive content.

For Task 2, the architecture based on a Bidirectional GRU combined with restricted self-attention showed the most promising results. It achieved a macro F1 score of 0.706, outperforming both the GCN-enhanced Transformer model and the custom Transformer variant. Notably, the custom Transformer model struggled with this task, reaching only a macro F1 score of 0.6094. The comparative advantage of the GRU-based model in this context indicates that sequential context modeling, when reinforced with a targeted attention mechanism, can be particularly effective for transfer learning scenarios where the model is pre-trained on hate speech data and fine-tuned on task-specific data.

In Task 3, which required the model to simultaneously detect the presence of label 1 and label 3 categories, the Transformer-GCN architecture again outperformed its GRU-based counterpart. Specifically, it achieved a macro F1 score of 0.65 for label 1 and an even higher 0.78 for label 3. This further strengthens the case for combining convolutional and transformer-based techniques when tackling nuanced, multi-label classification tasks, especially in multilingual settings.

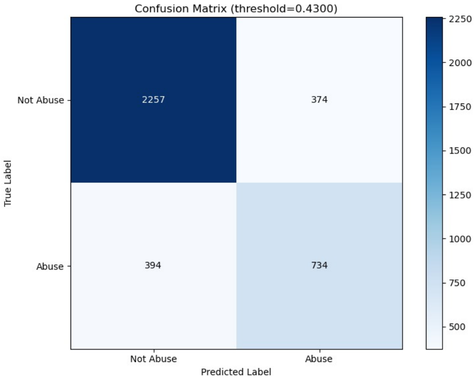

Figure 6: Confusion Matrix for Task 1

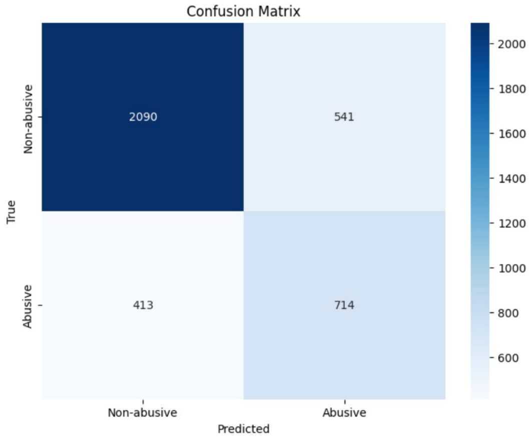

Figure 7: Confusion Matrix for Task 2

Overall, the results highlight the importance of architectural choices tailored to the nature of the task. While recurrent architectures like GRUs are beneficial in sequential learning scenarios, the combination of GCNs and Transformers appears to offer a more robust solution in tasks requiring complex feature extraction and cross-lingual generalization.

7 Conclusion

This paper presented our approach and results for detecting gendered abuse in online content. Our model using transformers, GCNs, and GRUs with mBERT embeddings achieved significantly higher macro F1 scores compared to baselines like custom transformers and CNN-BiLSTMs. The models effectively captured nuanced abusive language through localized feature learning and sequence modeling. Our analysis showed the impact of fac-

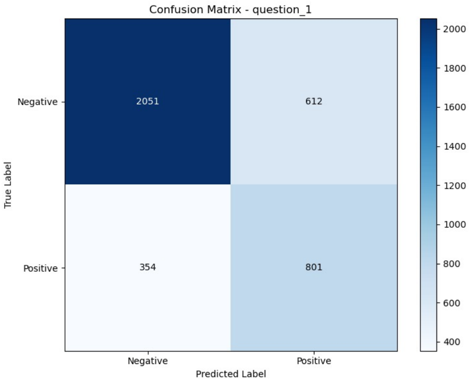

Figure 8: Confusion Matrix for Task 3, Label 1

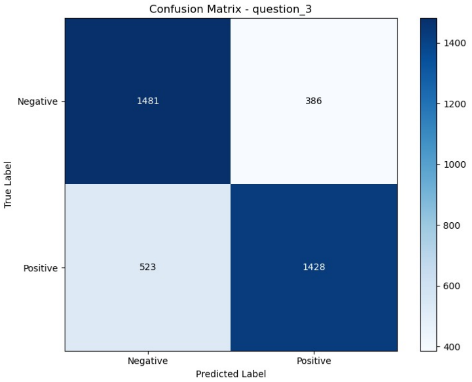

Figure 9: Confusion Matrix for Task 3, Label 3

tors like embedding techniques and input preprocessing on performance outcomes.

There remains difficulty in handling heavily code-switched languages—an area for future work. Through this shared task, we developed performant models for a crucial problem that limits online freedom of expression, particularly for marginalized gender identities. These models represent an important step toward creating safer, more inclusive digital spaces across Indic language communities.