Gendered Abuse Detection in Indic Languages

Dushyant Singh dushyant22181@iiitd.ac.in
Sujal Soni sujal22513@iiitd.ac.in
Mridul Goel mridul22303@iiitd.ac.in

1 Introduction

In today's digital age, the spread of abusive language on social media and other online platforms has become increasingly common. One particularly concerning form of this is gendered abuse, which targets individuals based on their gender and creates an unfriendly and often unsafe online environment, especially in communities that use Indic languages. This project was developed in response to the lack of reliable tools that can automatically detect such abuse. Gendered abuse is often subtle and depends heavily on context, which makes it difficult for traditional hate speech filters to identify. For instance, a message containing gender-based insults or offensive remarks might not be flagged by standard systems, even though it can deeply affect the individuals targeted. To address this issue, the project uses advanced natural language processing methods and transfer learning to accurately identify gendered abuse across various Indic languages.

2 Related Work

Over the past few years, there has been steady progress in the field of hate speech detection. Early methods mostly relied on fixed rules or basic statistical techniques, but these approaches often struggled to understand the subtle and complex nature of abusive language. More recently, researchers have turned to deep learning models—such as CNNs and LSTMs—which are better at capturing the meaning and flow of language in context. In particular, modern transformer-based models like BERT and XLM-R have shown strong results in detecting hate speech across multiple languages, making them well-suited for multilingual settings like ours. A recent study by Das et al. (2022) proposed data bootstrapping techniques to improve abusive language detection in low-resource Indic languages. They utilized models like m-BERT and MuRIL across 10 language settings, including code-mixed variants, and demonstrated significant performance gains. Their publicly released models and datasets have become valuable resources for multilingual abuse detection research. The ULI dataset used in this project, provides detailed labeling guidelines and benchmark results, offering a strong starting point for developing more advanced models. Additionally, previous work on identifying explicit language and using transfer learning to improve detection accuracy has played an important role in shaping the methods used in our study.

3 Dataset

We used the **ULI dataset**[1] for detecting and analyzing gendered abuse across three languages: **Hindi**, **Tamil**, and **Indian English**. This dataset was annotated by eighteen activists and researchers who have either experienced or studied gendered abuse online.

The dataset includes:

- 7,638 posts in English
- 7,714 posts in Hindi
- 7,914 posts in Tamil

Each language-specific corpus is divided into `train` and `test` splits.

Annotation was performed using three specific labeling questions, resulting in the following labels per post:

- **Label 1:** Is this post gendered abuse *when not directed* at a person of marginalized gender and sexuality?
- **Label 2:** Is this post gendered abuse *when directed* at a person of marginalized gender and sexuality?
- **Label 3:** Is this post explicit or aggressive in nature?

[1] https://github.com/tattle-made/uli_dataset

These multi-dimensional annotations enable comprehensive modeling of online abuse, particularly as it relates to marginalized identities.

4 Methodology

4.1 Data Loading and Label Aggregation

The input data was loaded from a labeled CSV file from ULI dataset, which consists of text posts annotated across six columns. Each column represents an individual annotator's judgment on whether the post is abusive or not. The format given above was same for the Hindi and Tamil languages as well.

Missing label values are safely handled using `NaN`-aware logic during label assignment.

4.2 Data Preprocessing

To ensure consistency and reduce noise in the textual data, the following preprocessing steps were applied:

- Removal of URLs, HTML tags, punctuation, newline characters, and numerical digits to eliminate irrelevant or noisy tokens from the text.
- Emoji normalization using the `emoji` Python library, which converts emojis into descriptive text tokens.
- Replacement of special characters, such as HTML-encoded entities like `&`, with their textual equivalents or removed as necessary.
- Lowercasing: All text was converted to lowercase for consistency.
- Tokenization: Tokenized using the pretrained XLM-R tokenizer, which ensures multilingual compatibility and subword handling.

These steps helped standardize the input format and improved model generalization by minimizing variability in the textual data.

4.3 Model Architecture

The proposed model architecture is designed for binary classification of abusive text for all languages. It uses a powerful multilingual language model called XLM-RoBERTa, which helps the system understand the meaning of words in context, even across different languages. To further improve understanding, the model includes Bidirectional GRU layers that look at the text both forward and backward and also focus on important parts of the sentence through an attention mechanism. These features help the model capture subtle cues that are often present in abusive or harmful language. The final layers simplify the information and make a prediction classifying either whether the post is abusive or how explicit it is. This approach balances deep language understanding with simplicity, making it well-suited for detecting complex forms of online abuse in multiple Indian languages. The key components of the architecture are described below:

- **Input Layer:** The model takes as input a sequence of token IDs and an attention mask, each of fixed length L which is different for each language.
- **XLM-RoBERTa Embeddings:** The input is passed through a pretrained `XLM-RoBERTa-Base` model from the HuggingFace Transformers library. This model is used to generate contextual embeddings for each token. The embedding layer is implemented using the `TFXLMRobertaModel` class.
- **Bidirectional GRU Layer:** The output embeddings are then fed into a **Bidirectional GRU** layer with 512 units in each direction. This allows the model to capture sequential dependencies in both forward and backward directions, improving contextual understanding of the sentence.
- **Multi-Head Attention:** The GRU output is passed through a **Multi-Head Self-Attention** mechanism with 4 attention heads and a key dimension of 64. This layer enables the model to focus on different parts of the sequence simultaneously, enriching the representation with global context.
- **Global Average Pooling:** To reduce the variable-length sequence output to a fixed-size representation, a `GlobalAveragePooling1D` layer is applied over the attention outputs.
- **Dropout Layer:** A dropout layer with a dropout rate of 0.2 is added to prevent overfitting by randomly deactivating neurons during training.
- **Output Layer:** The final layer is a fully connected `Dense` layer with a `sigmoid` activation for binary classification (Task-1 & 2) and

`softmax` for multi-class classification (Task-3). This produces a single scalar value between 0 and 1, which represents the probability of the input being abusive.

The model is trained using the `Adam` optimizer with a learning rate of 2×10^{-5}, and `binary cross-entropy` as the loss function. To address class imbalance, class weights are assigned accordingly to the non-abusive and abusive classes.

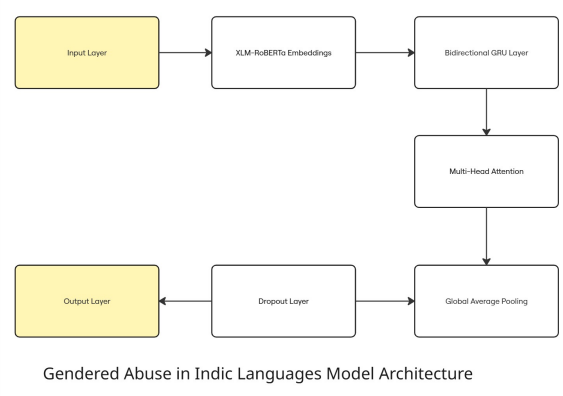

Figure 1: Model Architecture

4.4 Task-Wise Implementation Overview

This section talks about the specific details and architecture changes done in each section to complete the given task.

Task 1: Binary Classification to detect gendered abuse

Objective: In this task, the goal was to perform binary classification using only the provided ULI dataset to determine whether a given post contains gendered abuse. The model was trained on preprocessed data across all three languages—English, Hindi, and Tamil by following the steps outlined in Section 4.2. Labels were generated by aggregating the six annotator responses per post, with the most frequently occurring value selected as the final label.

Architecture: The architecture used for classification was consistent with the model described in Section 4.3, and a sigmoid activation was applied in the output layer to produce binary predictions. Model performance was evaluated using standard metrics such as precision, recall, and F1-score.

Task 2: Use Transfer Learning

Objective: In this task, the objective was to improve the model's ability to detect gendered abuse by leveraging external hate speech datasets through transfer learning. The approach involved a two-stage training process applied to English, Hindi, and Tamil data. The dataset used are- English[2], Hindi[3], Tamil[4]

Architecture: In the first stage, the model described in Section 4.3 comprising XLM-R embeddings, BiGRU, and Multi-Head Attention which was pretrained on publicly available Indic hate speech datasets using a sigmoid-activated output layer and minor architectural adjustments. In the second stage, this pretrained model was fine-tuned on the target dataset specific to each language. Early stopping was employed to prevent overfitting during fine-tuning. Then results were computed on the respective test datasets.

This method helped the model generalize better to real-world abusive content and significantly improved recall, especially for underrepresented abusive instances.

Task 3: Multi-Class Classification (Gendered + Explicit Language)

Objective: In this task, the objective was to build a multi-task classifier that jointly predicts both gendered abuse label_1 and explicit language label_3.

To generate a multi-class label for each instance in Task-3, two distinct annotation sources were utilized:

- `label_1` - Derived from six annotation columns in the from the Label-1 training files and aggregated using the maximum frequency value across annotators.
- `label_3` — Similarly computed the annotators columns in the Label-3 training files and aggregated using the maximum frequency value across annotators.

These were combined into a single unified label (`label`) using the following logic:

[2] https://github.com/t-davidson/hate-speech-and-offensive-language/blob/master/data/labeled_data.csv

[3] https://www.kaggle.com/datasets/iamtheoneaj/hindi-hate-speech-multi-labeled

[4] https://github.com/dreamspace-academy/ai-tamil-hate-speech-project/blob/master/dataset/dataset.csv

- **Label 0**: Neither `label_1` nor `label_3` are positive.
- **Label 1**: Only `label_1` is positive (only gendered abuse).
- **Label 2**: Only `label_3` is positive (only explicit language).
- **Label 3**: Both `label_1` and `label_3` are positive.(Both gendered abuse and explicit language)

This encoding enables the model to handle multi-label abusive behavior classification by distinguishing between different types of abuse: gender-based, explicit language, both, or none.

Architecture: The model architecture remained the same as used in previous tasks, comprising XLM-R embeddings followed by a Bidirectional GRU layer, Multi-Head Self-Attention, and Global Average Pooling. However, the final output layer was modified to use softmax activation which allows the model to predict one of four mutually exclusive classes. The model was trained for five epochs using categorical cross-entropy as the loss function and applied class weighting to address imbalances across the different classes.

4.5 Evaluation Metric

The primary evaluation metric used for all tasks is the **F1-Score**.

This metric is particularly suitable for imbalanced classification settings, as it ensures that minority classes (e.g., abusive categories) are not overshadowed by the majority class. It provides a balanced measure of precision and recall performance across all classes.

To address the challenge of class imbalance, especially where abusive instances are less frequent, we determined an optimal classification threshold using precision-recall curve analysis. Rather than using a fixed threshold of 0.5, we systematically evaluated the model's predictions across a range of threshold values. Precision and recall were computed at each point, and the threshold that achieved the highest F1-score—offering the best balance between identifying abusive posts and avoiding false positives—was selected for final classification. This strategy helped improve the model's ability to detect abuse more reliably.

5 Results

Based on the results in Tables 1–3, Baseline 2 achieved the highest Macro F1 scores for Task 1 in English (0.85), Tamil (0.85), and Hindi (0.78). For Task 2, XLM-BiGRU performed best on English (0.72), while Baseline 1 outperformed others on Hindi (0.69) and Tamil (0.78). In Task 3, XLM-BiGRU achieved the best performance across all three languages, with scores of (0.43) on English, (0.40) on Hindi, and (0.42) on Tamil.

Language	T1	T2	T3
English	0.68	0.53	0.20
Hindi	0.60	**0.69**	0.20
Tamil	0.79	**0.78**	0.21

Table 1: Macro F1 score of Baseline 1 across English, Hindi, and Tamil on Tasks 1, 2, and 3

Language	T1	T2	T3
English	**0.85**	0.65	0.32
Hindi	**0.78**	0.65	0.27
Tamil	**0.85**	0.75	0.39

Table 2: Macro F1 score of Baseline 2 across English, Hindi, and Tamil on Tasks 1, 2, and 3

Language	T1	T2	T3
English	0.66	**0.72**	**0.43**
Hindi	0.59	0.45	**0.40**
Tamil	0.78	0.53	**0.42**

Table 3: Macro F1 score of XLM-BiGRU across English, Hindi, and Tamil on Tasks 1, 2, and 3

6 Analysis and Observations

Our model demonstrated slightly lower F1 scores compared to the baseline models across task 1 and 2. This performance gap can largely be attributed to the limited quantity of labeled training data, which constrained the model's ability to generalize effectively. We employed a hybrid architecture combining the multilingual **XLM-RoBERTa** model with a **BiGRU** layer and a **multi-head self-attention** mechanism. While XLM-RoBERTa provided robust language representations, the subsequent BiGRU layer captured sequential dependencies, and the attention module enhanced focus on relevant features. However, despite the architecture's strength, data sparsity and class imbalance

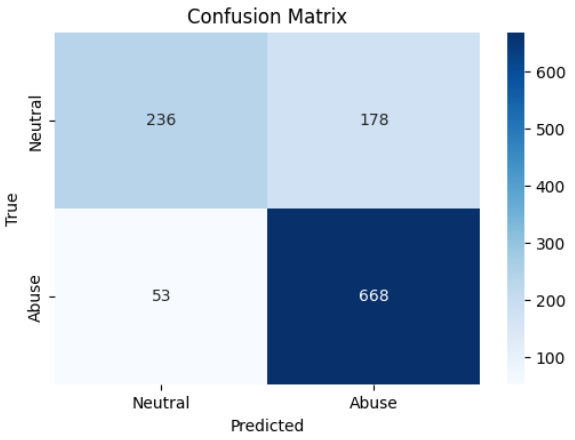

Figure 2: Confusion Matrix for Task-1 on tamil test set.

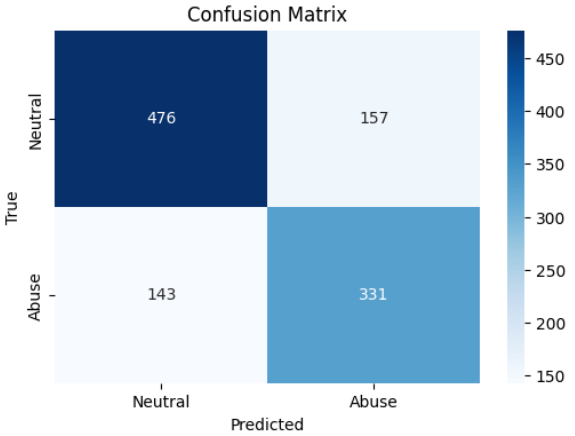

Figure 3: Confusion Matrix for Task-2 on english test set.

remained key challenges. To maintain training efficiency across all language-task combinations, we opted for a relatively lightweight setup. We anticipate that with access to more annotated data and further tuning of hyperparameters, the model's performance can be significantly enhanced in future work.

7 Conclusion and Future Work

We introduced a multilingual model architecture that combines **XLM-RoBERTa**, a **BiGRU** layer, and **multi-head self-attention** to address diverse language classification tasks. The model balances performance with efficiency, making it suitable for multilingual and multi-task settings with limited resources.

Future work will focus on enhancing performance through two key directions: (1) expanding the training dataset using data augmentation or semi-supervised learning to address sparsity and

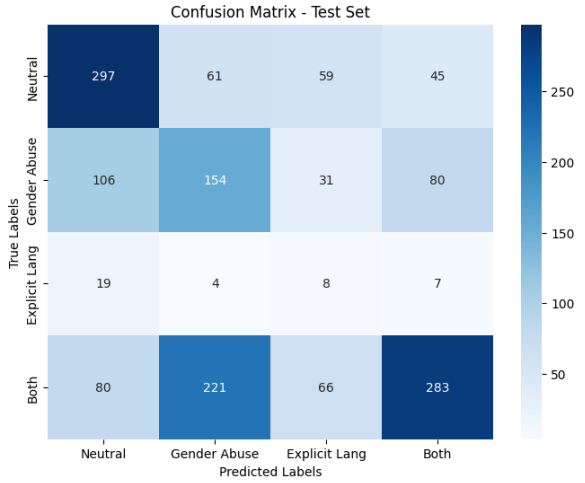

Figure 4: Confusion Matrix for Task-3 on hindi test set.

imbalance, and (2) refining the model via targeted hyperparameter tuning and task-specific optimization. These improvements aim to boost generalization across languages and tasks in future iterations.

References

[1] *Breaking the Silence Detecting and Mitigating Gendered Abuse in Hindi, Tamil, and Indian English Online Spaces*. https://arxiv.org/abs/2404.02013

[2] *ULI Dataset*. https://github.com/tattle-made/uli_dataset

[3] *Refrenced CNN BiLstm Model Code Repository* https://github.com/advaithavetagiri/CNLP-NITS-PP

[4] *Data Bootstrapping Approaches to Improve Low Resource Abusive Language Detection for Indic Languages*. https://arxiv.org/abs/2204.12543

[5] *Breaking the Silence: Detecting and Mitigating Gendered Abuse in Hindi, Tamil, and Indian English Online Spaces*. https://arxiv.org/abs/2404.02013

[6] *Hate Speech and Offensive Language Dataset (English)* . https://github.com/t-davidson/hate-speech-and-offensive-language/blob/master/data/labeled_data.csv

[7] *Hindi Hate Speech Multi-Labeled Dataset (Kaggle)*. https://www.kaggle.com/datasets/iamtheoneaj/hindi-hate-speech-multi-labeled

[8] *AI Tamil Hate Speech Project – Dataset*. https://github.com/dreamspace-academy/ai-tamil-hate-speech-project/blob/master/dataset/dataset.csv