Project Proposal: Gendered Abuse Detection in Indic Languages

Praddume Attri **Prayag Parashar** **Tarandeep Singh**
praddume22358@iiitd.ac.in,prayag22377@iiitd.ac.in,tarandeep22536@iiitd.ac.in,

Abstract

Online gender-based violence exacerbates social and economic vulnerabilities, ranging from harassment to severe threats. Automated detection is crucial, We develop a language model to detect gender-based abuse in social media. Our dataset, sourced from Instagram and Twitter comments

1 Introduction

In the digital age, social media platforms such as Instagram and Twitter have become essential tools for communication, self-expression, and community building. However, these platforms also serve as breeding grounds for online gender-based violence (GBV), which includes a spectrum of harmful behaviors—from subtle harassment and demeaning comments to explicit threats and abuse. This pervasive issue disproportionately affects women and gender minorities, particularly in under-resourced linguistic communities, amplifying their social and economic vulnerabilities.

For instance, a female entrepreneur promoting her startup on Twitter might receive replies like "Stick to the kitchen" or "Women can't run businesses." Similarly, a woman posting a celebratory photo on Instagram might be bombarded with comments like "This is why girls get harassed" or "Attention-seeker, go cover yourself." These remarks not only perpetuate harmful gender stereotypes but also silence voices and discourage participation in online spaces.

Manual moderation of such content is neither scalable nor consistent, especially across multiple languages. The urgency to address this challenge is heightened in countries like India, where internet access is growing rapidly, yet content moderation in Indic languages remains limited. Many forms of gender-based abuse go undetected due to the lack of annotated data and linguistic tools in regional languages such as Hindi and Tamil.

To tackle this issue, our project focuses on developing an automated, multilingual language model capable of detecting gendered abuse across English, Hindi, and Tamil. By leveraging annotated social media comments from Instagram and Twitter, we aim to build a system that not only identifies toxic behavior but also contributes to safer and more inclusive online spaces for all users.

2 Related Work

Prior research in online abuse detection has primarily focused on English-language datasets, with models trained to identify hate speech, toxic comments, and cyberbullying. Notable efforts include the use of transformer-based models like BERT and RoBERTa for toxic comment classification on platforms such as Reddit, YouTube, and Twitter.

Studies such as Davidson et al. (2017) proposed hate speech detection using logistic regression on annotated tweets, while Waseem and Hovy (2016) highlighted the importance of socio-linguistic context in labeling abusive language. More recent work has leveraged multilingual models like XLM-R for cross-lingual transfer in abuse detection, but performance drops significantly when applied to under-resourced languages.

In the Indian context, very few annotated datasets exist for gender-based violence in Indic languages. Some initial efforts, like the HASOC and FIRE shared tasks, have explored hate speech in Hindi and Tamil, but they lack a focused lens on gendered abuse. Our work aims to fill this gap by creating a targeted, annotated dataset in English, Hindi, and Tamil, and developing a multilingual model specifically for detecting gender-based abuse.

3 Methodology

3.1 Preprocessing

Tokenization removes punctuation, handles emojis, expands contractions/acronyms, splits hashtags, and normalizes text. CSV datasets are processed by extracting and averaging labels and structuring data.
Dataset Used: The dataset was loaded from the specified CSV file and preprocessed for model training.
Handling Missing Values: Any missing values were either filled using mean imputation or dropped if deemed insignificant.

2. Handling Class Imbalance via Data Augmentation
The dataset displayed class imbalance, which can bias model predictions. To address this:

Synonym Replacement Augmentation was applied using nlpaug's SynonymAug (WordNet-based) to generate new, semantically similar sentences for underrepresented classes.

Tweets from minority classes were randomly sampled and augmented to match the frequency of the majority class.

The resulting dataset was balanced, improving the model's ability to generalize across all classes.
Text Cleaning: Each tweet was cleaned by:
Replacing user mentions (@username) with a generic user token.
Removing HTML entities (e.g., amp;).
Removing URLs, special characters, and unnecessary punctuation. Feature Scaling: StandardScaler was used to normalize the feature columns to ensure uniform scaling, which is essential for Elastic-Net regularization to perform optimally.
Train-Test Split: The dataset was split into 80Data Inspection: Target variable distribution and feature correlations were visually and statistically examined to understand data trends.

3.2 Text Vectorization and Embedding

All tweets were tokenized and converted into sequences using Keras's `Tokenizer`. The sequences were padded to a uniform maximum length.

- FastText Word Embeddings
- Pre-trained 300-dimensional FastText embeddings (`wiki-news-300d-1M-subword.vec`) were loaded.
- An embedding matrix was constructed to map each word in the vocabulary to its corresponding FastText vector.
- Unknown words were initialized with small random values.

The embedding layer was initialized with these vectors and set to non-trainable to retain semantic information.

3.3 Model Architecture

A hybrid deep learning architecture was designed to capture both contextual and global information from tweets:

3.3.1 Embedding Layer

The embedding layer was initialized with Muril embedding vectors (non-trainable).

3.3.2 Bidirectional LSTM

A Bidirectional LSTM layer was used to capture both forward and backward context of each word in the sequence, with dropout regularization applied to avoid overfitting.

3.3.3 Multi-Head Attention

The multi-head attention mechanism was employed to add richer attention features and learn multiple representation subspaces for the same input.

3.3.4 Global Average Pooling

A global average pooling layer was used to reduce the sequence output to a fixed-size context vector, allowing for efficient feature extraction.

3.3.5 Dense Layers

Two fully connected dense layers were used, each with ReLU activation, dropout, and L2 regularization to prevent overfitting.

3.3.6 Output Layer

The output layer consisted of a softmax activation function for 3-class classification, producing probabilities for each class.

This architecture balances complexity with regularization to minimize overfitting.

3.4 Training Strategy

3.4.1 Loss Function

The loss function used was `categorical_crossentropy`, which is suitable for multiclass classification problems.

3.4.2 Optimizer

The `Adam` optimizer with the default learning rate was used for efficient convergence during training.

3.4.3 Class Weights

Class weights were automatically computed to balance the influence of each class during training, ensuring that underrepresented classes received appropriate emphasis.

3.4.4 Early Stopping

Early stopping was implemented by monitoring validation accuracy. Training was stopped early if no improvement was observed for 3 consecutive epochs, preventing overfitting and saving computational resources.

3.5 Evaluation Metrics

The performance of the model was evaluated using the following metrics:

3.5.1 Accuracy

Accuracy was computed on the validation data to measure the overall performance of the model.

3.5.2 Confusion Matrix

A confusion matrix was used (optional for analysis) to provide insights into the model's ability to distinguish between the classes.

3.5.3 Precision, Recall, F1-score

Precision, recall, and F1-score were computed for each class (optional for class-wise insights) to assess the model's performance in terms of its ability to correctly identify each class and balance false positives and false negatives.

4 Dataset Description

The dataset includes 7,638 English, 7,714 Hindi, and 7,914 Tamil posts. Each post is annotated for (1) gendered abuse in general, (2) gendered abuse toward marginalized groups, and (3) explicit/aggressive content. Labels include `1` (matches), `0` (does not match), `NL` (not annotated), and `NaN` (not assigned).

The dataset (`train.csv`) contains each tweet three times (one per label) with columns for `id`, `text`, `language`, `key` (label type), and annotator responses for all three languages.

Dataset 2 Overview

Source: The dataset comprises tweets collected using the Twitter API, focusing on English-language content.

Size: Approximately 24,783 tweets.

Annotations: Each tweet is labeled by multiple annotators into one of the following three categories:

- Hate Speech
- Offensive Language
- Neither

The final label for each tweet is determined by a majority vote among the annotators.

Format: The dataset is structured in a CSV file with the following columns:

count Number of annotators who labeled the tweet
hate_speech_annotation Number of annotators who labeled the tweet as hate speech
offensive_language_annotation Number of annotators who labeled the tweet as offensive language
neither_annotation Number of annotators who labeled the tweet as neither
label Final label based on majority vote (0 = Hate Speech, 1 = Offensive Language, 2 = Neither)
tweet The tweet text

Results/Findings

- **LSTM**: Achieved moderate performance with an F1 score, precision, and recall metrics. The confusion matrix highlighted challenges in distinguishing between "hate" and "offensive" classes.
- **BERT**: Demonstrated strong performance with high validation accuracy (up to $\sim$83%) after 3 epochs, indicating its effectiveness in contextual understanding. Training plots showed stable convergence.
- **Bidirectional LSTM with Muril embeddings + Attention**: Addressed class imbalance using synonym-based augmentation. Achieved $\sim$81% validation accuracy, with improved performance on minority classes after augmentation.

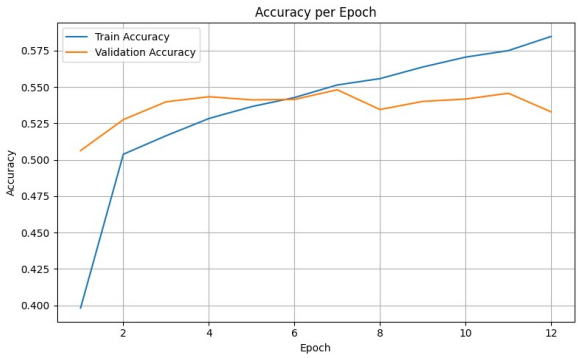

Figure 1: bidirectional lstm with attention

Classification Report:

	precision	recall	f1-score	support
hate	0.33	0.67	0.44	290
offensive	0.97	0.88	0.92	3832
neither	0.84	0.89	0.87	835
accuracy			0.87	4957
macro avg	0.71	0.81	0.74	4957
weighted avg	0.91	0.87	0.89	4957

Figure 2: Hate Speech

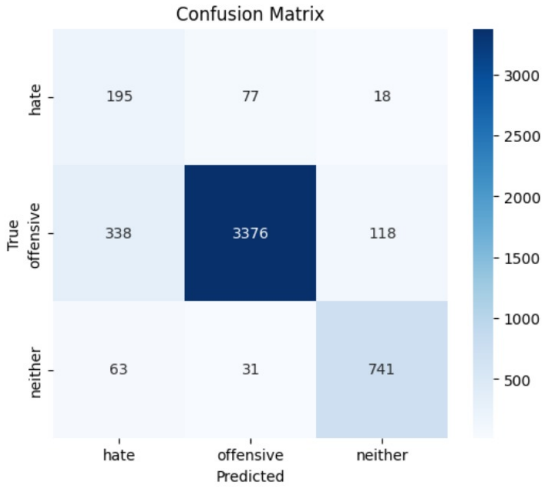

Figure 3: Confusion Matrix

Key Metrics

- All models prioritized reducing false positives/negatives for "hate" and "offensive" classes.
- BERT outperformed LSTM and BiLSTM+Attention in validation accuracy, suggesting transformer architectures are better suited for nuanced language tasks.

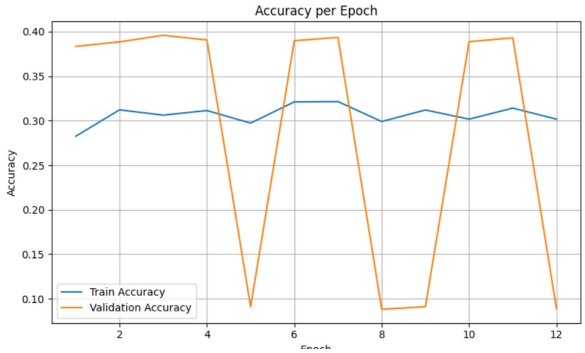

Figure 4: simplernn

Data Insights

- Class imbalance in the dataset (e.g., "offensive" class dominance) was mitigated via augmentation, improving model robustness.
- Preprocessing steps (e.g., removing noise, stopwords) were critical for model performance.

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
BiLSTMAttentionClassifier                [32, 3]                   --
├─LSTM: 1-1                              [32, 50, 256]             919,552
├─Linear: 1-2                            [32, 50, 1]               257
├─LayerNorm: 1-3                         [32, 256]                 512
├─Linear: 1-4                            [32, 3]                   771
├─Sigmoid: 1-5                           [32, 3]                   --
==========================================================================================
Total params: 921,092
Trainable params: 921,092
Non-trainable params: 0
Total mult-adds (Units.GIGABYTES): 1.47
==========================================================================================
Input size (MB): 4.92
Forward/backward pass size (MB): 3.36
Params size (MB): 3.68
Estimated Total Size (MB): 11.96
==========================================================================================
```

Figure 5: Lstm

5 Future Work

- Experiment with advanced transformer variants (e.g., RoBERTa, GPT-3) or hybrid architectures.
- Incorporate cross-validation and hyperparameter tuning for optimization.

6 Data Improvements

- Expand datasets to include multilingual and dialect-specific gendered abuse examples.
- Explore advanced augmentation techniques (e.g., back-translation, GPT-based paraphrasing).

7 Deployment & Robustness

- Test real-time deployment in social media moderation systems.
- Evaluate robustness against adversarial attacks and sarcasm.

8 Conclusion

The notebook successfully implements LSTM, BERT, and BiLSTM+Attention models for gendered abuse detection. BERT's superior performance underscores the value of pre-trained contextual embeddings. Challenges like class imbalance and generalization to unseen data remain. Future efforts should focus on refining models, expanding datasets, and practical deployment to enhance societal impact.

9 References

https://arxiv.org/abs/2311.09086
https://www.sciencedirect.com/science/article/pii/S2590005622000625