AniMOJity: Detecting Hate Comments in Indic languages and Analysing Bias against Content Creators

Rahul Khurana[*], **Chaitanya Pandey**[*], **Priyanshi Gupta**[*], **Preeti Nagrath**
Department of Computer Science and Engineering,
Bharati Vidyapeeth's College of Engineering,
New Delhi, India
{rahulkhurana.rk59, chaitanya.p2001, guppriyanshi, preetinagrath1}@gmail.com

Abstract

Online platforms have dramatically changed how people communicate with one another, resulting in a 467 million increase in the number of Indians actively exchanging and distributing social data. This caused an unexpected rise in harmful, racially, sexually, and religiously biased Internet content humans cannot control. As a result, there is an urgent need to research automated computational strategies for identifying hostile content in academic forums. This paper presents our learning pipeline and novel model, which classifies a multilingual text with a test f1-Score of 88.6 % on the Moj Multilingual Abusive Comment Identification dataset for hate speech detection in thirteen Indian regional languages. Our model, Animojity, incorporates transfer learning and SOTA pre- and post-processing techniques. We manually annotate 300 samples to investigate bias and provide insight into the hate towards creators.

1 Introduction

With the unbridled spread of Internet culture, hopes of finding an accepting community have led many racially, culturally, and sexually diverse groups to take refuge in their corner of the Internet, showcasing the strength of online forums. However, those seeking to spread hateful content look for ways to circumvent the restrictions placed on social media and hinder their healthy development. Due to the societal concern hate speech has garnered, there is a strong motivation to make advancements in its automatic detection. Online hate speech in general, and gendered online hate speech in particular, have become an issue of growing concern in both social and professional discourses.

Before we delve into detecting hate speech, it is imperative to understand its definition clearly. (Ross et al., 2017) believes that a distinct definition of hate speech can make the annotation process easier, leading to reliable detection of what categorizes as offensive. Nevertheless, hate speech and appropriate free expression walk a fine line, making its definition not fully agreeable. We opt to build upon existing definitions laid down by (Davidson et al., 2017), (De Gibert et al., 2018), and (Fortuna and Nunes, 2018) instead of proposing a specific definition.

Another issue that is not brought to light as often is that users with a diverse linguitic backgrounds tend to switch between different languages while expressing their thoughts on social media, limiting the capabilities of a monolingual model and necessitating the need for multilingualism in a model. We address this challenge by building a novel model that detects hate comments for thirteen regional languages (Hindi, Urdu, Telegu, Marathi, Gujarati, Malayalam, Punjabi, Assamese, Kannada, Bengali, Tamil, Rajasthani, Haryanvi) using the Moj Multilingual Abusive Comment Identification dataset. While working with multilingual data, apart from a low resource issue, there is a tendency for imbalanced sample distribution. Given that there is a relatively lower number of samples categorized as hateful in less-used languages, it encourages us to adopt transfer learning, data augmentation, and other techniques in AniMOJity.

We base our study on detecting hateful comments for Indian regional languages and analyzing whether they have a biased perspective in the comments towards a particular community. Our main contributions are

- We propose a novel hate speech detection model in a low-resource Indic multilingual setting that incorporates transfer learning and documents the effect of different algorithms on our dataset.

- Our pipeline uses state-of-the-art post-processing techniques to handle hateful behavior by automatically flagging offensive posts.

[*] Equal contribution

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 172 - 182
December 15-18, 2022 ©2022 Association for Computational Linguistics

- By observing dominant topics (gender, clothing, age, religion, race) in the comments, we manually annotate 300 samples for these bias categories. This dataset broadens the scope of analysis research for social media platforms.
- We further provide a comprehensive analysis using LDA on our dataset to determine the specific keywords and perspectives of these commenters towards the content creators, who are at the brunt of this hate culture.

The remainder of the paper is structured as follows. Section 2 looks at some of the similar works in this domain. Section 3 describes the training dataset, followed by an in-depth presentation of our methodology incorporating data pre-processing and model architecture in Section 4. Next, in Sections 5 and 6, the experimental setup and results are explored, followed by Section 7, where we analyze bias in comments and provide a detailed overview of our findings. We conclude with section 8, discussing future avenues for our proposed model.

2 Related Work

This section briefly sheds light on the various methodologies adopted to tackle hate speech detection and multilingual text classification over the past few years, serving as a benchmark for our research.

Hate Speech Detection Hate speech detection has been at the center of the academic community's attention due to its constantly evolving and picking up different forms with time. The problem categorizes itself as binary or multi-class classification. (Waseem and Hovy, 2016) created a three-class Twitter dataset annotated as sexist, racist, and neutral for offensive language detection. (Kumar et al., 2018) showcased their findings on an aggression identification task discriminating 15,000 annotated Facebook posts and comments in English and Hindi as non-aggressive, covertly aggressive, and overly aggressive. (Davidson et al., 2017) presented a 24,000 corpus for identifying English tweets belonging to profanity, hate speech, and non-offensive categories. (Mandl et al., 2019) gave a detailed account of offensive language identification where three datasets available for Hindi, German, and English were created from Twitter and Facebook. (Zampieri et al., 2019) and (Zampieri et al., 2020) presented their results in several languages obtained from the SemEval competition.

Multilingual Text Classification Multilingual text classification (MTC) aims to breach the language barrier by improving monolingual models by scaling to different languages. (Prajapati et al., 2009) introduced the implementation of translating documents to a universal language for classification, which was bolstered by (Li et al., 2018) to extract grammatical and semantic features from the translated dataset before classification. However, the noise accumulated by translation errors creates a disparity in the final results. (Amini et al., 2010) combined two semi-supervised learning techniques, co-regularization, and consensus-based self-training, to investigate multilingual text classification on a dataset containing five different languages: English, German, French, Italian, and Spanish. (Mittal and Dhyani, 2015) studied MTC in Spanish, Italian, and English by using the N-gram technique and Naïve Bayes to predict the language of a document in classification. (Bentaallah and Malki, 2012) compared two wordnet-based approaches for multilingual text categorization. One relies on the WordNet associated with each language while excluding the translation, and the other focuses on a dis-ambiguation strategy to focus on the most common meaning of the word and access WordNet using a machine translation. Data augmentation ((Ibrahim et al., 2018)) and transfer learning (Roy et al., 2021) help combat situations where there is a lack of training data, both of which are adopted to improve our training data. Recently, promising techniques involving deep learning and contextual embeddings have spearheaded a dynamic shift in the approach to tackling MTC tasks. Transformers became a mainstay in cross-lingual tasks and ushered in mBERT (Devlin et al., 2018), a multilingual masked language model, and XLM (Conneau and Lample, 2019). (Khanuja et al., 2021) proposed a multilingual LM, MuRIL, specifically built for Indic languages.

Impact of Biased Attack on Social Media Biases Make People Vulnerable to Misinformation Spread by Social Media, and cognitive biases originate in how the brain processes the information every person encounters daily. The study by (Döring and Mohseni, 2020) analyses the comments on YouTube and displays a gender bias in the comments. Most attacks are against female content creators and are not just hateful but offensive. Furthermore, the analysis by (Aguirre and Domahidi, 2021) portrays that the comments on YouTube are sexual as well as racist in nature. Thus, biased and offensive comments against them can highly ruin their image. Our work combines pre and post-processing techniques with a novel transfer learning pipeline for hate speech detection in low-resource languages, as well as analyzes the bias in comments against content creators on the Moj platform.

3 Dataset

The dataset utilized in this study was made available by the Moj Multilingual Abusive Comment Identification Challenge organizers in partnership with IIIT-D as part of that challenge [1]. Given the natural language and contextual user data, the project aims to combat abusive comments on Moj, one of India's largest short-form video apps, in thirteen languages, as shown in Figure 1. Figure 2 shows the distribution of which language contains the most hateful comments.

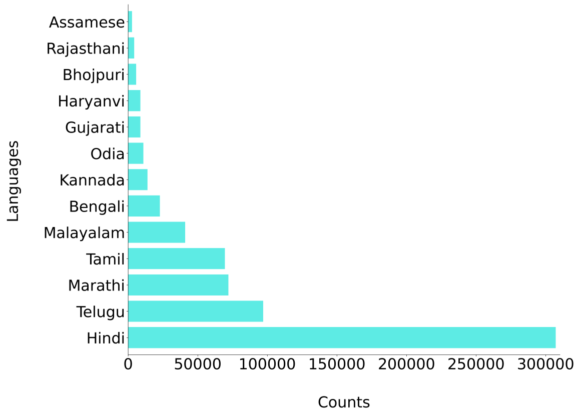

Figure 1: Distribution of languages present in the Moj Multilingual Abusive Comment Identification dataset

3.1 Data Pre-processing

Data preprocessing aims to maintain the input text's original grammatical structure and linguistic information while reducing stop-words, inhibiting the loss of information. To retain key information, we followed the following preprocessing steps

- We created a list of common stop-words to remove from the dataset for each language and converted the text to lowercase. Here stop-words are nugatory words that do not influence the output.
- We substituted emojis by their linguistic meaning in the tweet for each source language. To capture the contextual meaning of an emoji, we tried to incorporate emoji2vec. However, due to the diverse nature of our dataset, apart from a few languages like Hindi, there was not any pre-existing support for languages like Assamese, Gujrati, etc.

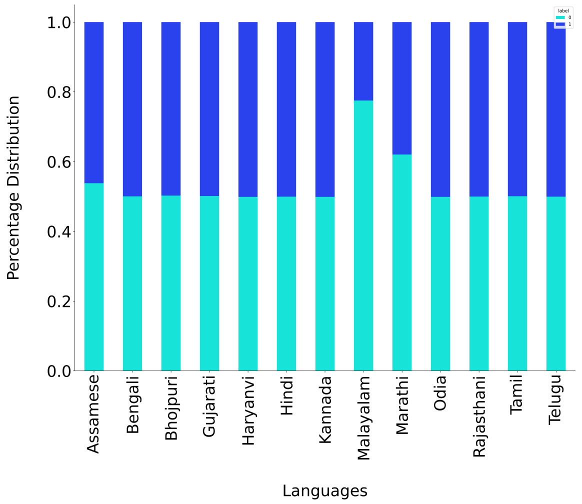

Figure 2: Percentage distribution of Hate vs Non-hate in each language in the Moj Multilingual Abusive Comment Identification dataset

The conventional method of using ekphrasis for preprocessing does not work well with a multilingual dataset, encouraging us to adopt Indic NLP (Kakwani et al., 2020) and NLTK library support for preprocessing Hindi. However, due to the tokenization constraints in Indic NLP, we perform tokenization using XLM-R.

4 Methodology

This section documents the techniques used to achieve the study's main objective.

4.1 XLM-R Model and finetuning

We use XLM-RoBERTa (XLM-R), a universal cross-lingual model trained on 100 different languages, using input ids to determine the language used. One of the critical differences XLM offers over its counterparts is the fact that it uses a stream of an arbitrary number of sentences, truncating the ones exceeding a limit. Unlike some XLM multilingual models, XLM-R does not require language tensors to identify the language used and can determine the correct language from the input id. We

[1] https://www.kaggle.com/competitions/iiitd-abuse-detection-challenge

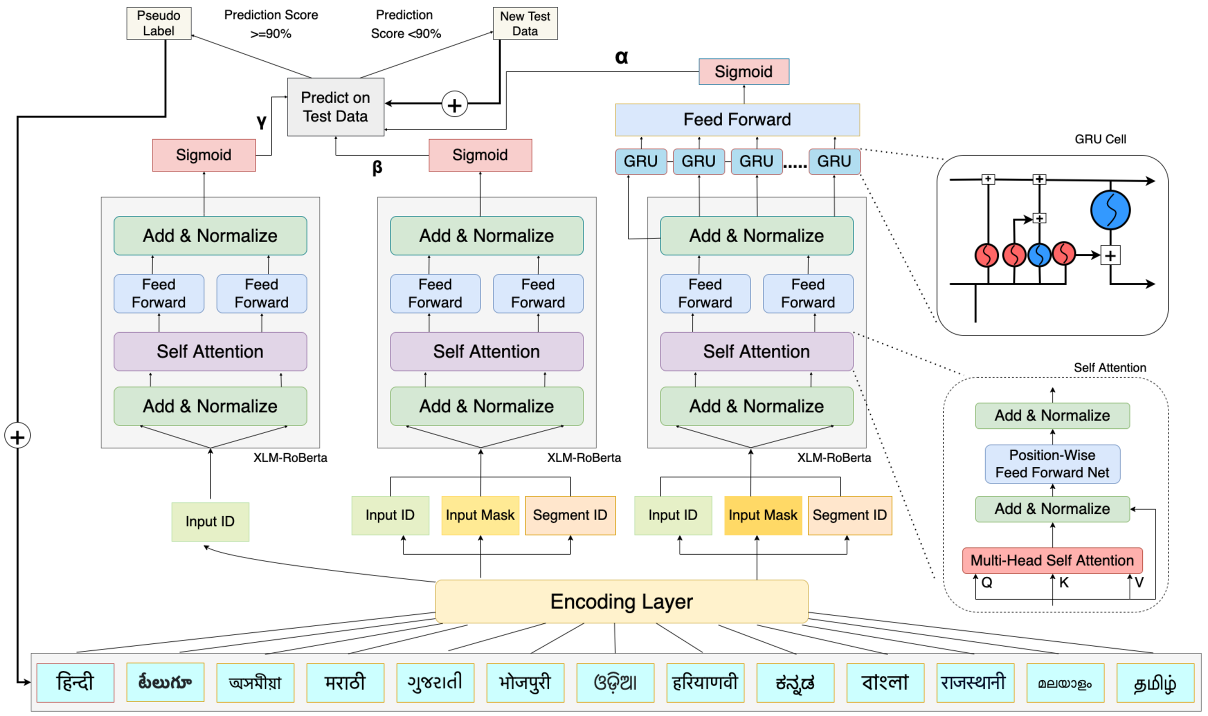

Figure 3: **Illustration of AniMOJity:** It consists of three language models merged together using α, β, γ as weighted parameters

fine-tuned our model by adding layers to the core model using pre-trained artificial neural networks.

4.2 Incorporating MLM with Fine Tuning

In Masked Language Modelling, a fixed percentage of words are masked, and the model is expected to predict the masked words based on the other words. During fine-tuning, the parameters of the pre-trained models are frozen while the detection layer is updated using an optimization algorithm to minimize the loss function.

$$\max_{\theta} \log p_{\theta}(\overline{\mathbf{x}} \mid \hat{\mathbf{x}}) \approx \sum_{t=1}^{T} m_t \log p_{\theta}\left(x_t \mid \hat{\mathbf{x}}\right) \qquad (1)$$

In equation 1, we maximize the probability of a masked token x_t to appear in the t ’th position in a sequence given the tokens in that sequence, x_hat.

4.3 Model Design

The architecture for AniMOJity (Figure 3) is described in this section. We used a combination of three distinct models as described below:

- In our first model, we pass Input-Id as the vectorized input through a pre-trained XLM-R model (Conneau and Lample, 2019). The output (last-hidden-state) obtained from the model is passed as an input to a Dense layer having sigmoid as an activation function and binary cross-entropy as the loss function. Adam optimizer with a learning rate of 5e-6 was used to train the model.

- For our second model, we pass Input-Id, token-type-id, and attention mask as the vectorized input through a pre-trained XLM-R model. The output (last-hidden-state) obtained from the model is passed as an input to the Dense layer having sigmoid as the activation function and binary cross-entropy as the loss function. Adam optimizer with a learning rate of 1e-5 is used to train the model.

- Finally, in the third model, we pass Input-Id, token-type-id, and attention mask as the vectorized input through a pre-trained XLM-R model. The output (last-hidden-state) obtained from the model is passed as an input to the GRU cell (Chung et al., 2014) (having 128 units), and output from the GRU cell is flattened and connected to a Dense Layer having sigmoid as the activation function and binary cross-entropy as the loss function. Adam optimizer with a learning rate of 1e-5 is used to train the model.

Algorithm 1: AniMOJity's Training Algorithm

```
Data: Hateful comment dataset
Result: AniMOJity: A meta-model used to predict hateful comments
Create 3 XLM-R-based Models;
  while $I \neq 2$ do
      while $J \neq 3$ do
          Train Model-(j) on Training Dataset and record the result for Test dataset;
            $J \leftarrow J + 1$;
      end
      Predictions on test set (Y-hat) = $\alpha$ * (Predictions from Model-(1)) + $\beta$ * (Predictions from Model-(2)) + $\gamma$ *
        (Predictions from Model-(3));
        Y-hat (having a confidence score above 90% for hateful and below 10% for not hateful comments) are used as
        features to create an augmented dataset for the (ith)-level model;
        $I \leftarrow I + 1$;
end
the (2)-level model makes predictions on the test set
```

For model stacking, we combine their predictions to create a model using the fusion weights (α, β, γ) shown in Figure 3. We train multiple base models to predict a target variable while concurrently using the predictions of each model to predict the value of the target variable.

4.4 Psuedo labelling

Pseudo-labeling involves using labeled data to predict unlabelled data. The trained model generates pseudo labels for an unlabelled dataset, combined with the original labels for a final model training. This improves the model's robustness by creating a more precise decision boundary. We implement pseudo-labeling while working on our dataset by utilizing the labels where the predictions on the test set have a confidence score of more than 90% and less than 10% for hateful and inoffensive comments, respectively.

4.5 Text Classification

During text classification, a transformer model takes the final hidden state (h) of the first token [CLS] as the representation of the whole sequence. To classify a comment as hateful or not, we pass the fine-tuned representation of the comment to a sigmoid function (equation 2) and train the model to optimize the binary cross entropy loss (equation 3).

$$\text{p}(\text{c} \mid \mathbf{h}) = softmax(\text{Wh}) \tag{2}$$

Here, W denotes the weights for the classification layer and h is the final hidden state.

$$L_{\text{BCE}} = -\frac{1}{m}\sum_{i=1}^{m}\left(y_{(i)} \log\left(\hat{y}_{(i)}\right) + \left(1 - y_{(i)}\right)\log\left(1 - \hat{y}_{(i)}\right)\right) \tag{3}$$

Where, $\text{y}_{(i)}$ and y_t represent the ground truth and predicted class of the i^{th} sample in a dataset. Since binary classification means a class takes either 0 or 1 as its input, if $y_{(i)} = 0$ term ceases to exist, and if $y_{(i)} = 1$ then the $\left(1 - y_{(i)}\right)$ term becomes 0.

5 Experimental Setup

This section elaborates on the baselines, modules, and functions used to construct AniMOJity. As discussed in related work, we used m-BERT (Devlin et al., 2018) as our baseline model, which has been the golden standard for multilingual text classification tasks. Using m-BERT as our backbone, we experimented with different architectures, among which the instances where we found promising results are seen in 1. However, the primary limitation with m-BERT was the lack of support for low-resource languages, which led us towards MuRIL (Khanuja et al., 2021) since it has been trained on a wide assortment of Indian regional languages to improve downstream NLP tasks. However, the most significant update XLM-Roberta offers over a model confined to a limited amount of training data, like MuRIL, is the significantly increased amount of training data, which in conjunction with the Masked Language Modelling approach discussed in Section 4.2, cements it as state of the art.

We first split our task into two pipelines: learning and testing, where the training (learning) process is carried out twice before making predictions on the test set as explained in Algorithm 1 below. We employ two models showing state-of-the-art results on multilingual classification, mBERT, and XLM-R (Conneau and Lample, 2019) as baselines for

Table 1: Performance evaluation of the binary hate speech classification based on Moj Multilingual Dataset for 13 low resource languages in terms of Accuracy and F1 Score

S.No.	Model	Accuracy (%)	Test F1 Score (%)
1	CNN - Single Input (m-Bert)	93.269	87.181
2	CNN - Multiple Input (m-Bert)	93.420	87.449
3	CNN (m-Bert)	92.898	87.232
4	Concat Pooling	94.726	87.933
5	MuRIL	93.30	87.420
6	CNN - Single Input (XLM-R)	93.823	87.619
7	CNN - Multiple Input (XLM-R)	94.333	88.369
8	GRU Cell (XLM-R)	94.628	88.377
9	CNN - Attention (XLM-R)	94.529	88.497
10	CNN - LSTM (XLM-R)	94.240	88.290
11	CNN – BiLSTM (XLM-R)	**96.324**	87.181
12	**AniMOJity**	95.604	**88.602**

Table 2: Hate and Bias-level breakdown of the multi-label 300 annotated samples. Note that we can observe more than one category of bias for a comment

Type	Gender	Clothing	Age	Racial	Religion
Hate	72	148	30	82	65
No Hate	102	26	140	92	109

our binary classification task. While working on these transformer-based models, we came across three types of inputs: input-id, token type id, and attention mask, of which we tested different combinations. The output (last-hidden-state) obtained from these models is passed as an input to the classification head. The models implemented in this study are created using Python 3.10.0 with Tensorflow v2.6.1 as the deep learning framework to build the architecture and train on Graphical Processing Unit (GPU Tesla P100 16GB) servers of Kaggle. To evaluate the performance of our system, we conduct experiments comparing different models. We created and trained our models using TensorFlow and Keras after dividing the dataset into a 90/10 ratio. Accuracy and F1 scores are used as the evaluation criteria. We assess the outcomes of the architecture shown in Table 1.

6 Results

In this section, we describe the evaluation results obtained after testing each language and briefly examine the performance of different models.

Practical Findings By comparing the proposed model with benchmarks, we demonstrate the effectiveness of our architecture. Table 1 shows the investigative analysis of different strategies we used on the multilingual task. After exhaustive experimentation with different architectures, we exhibit the capability of AniMOJity to deal with offensive language detection.

Theoretical Findings Our suggested methodology performed very well when applied to comments where a nasty word or phrase guided the user's intent after using AniMOJity to categorize remarks as offensive or inoffensive. There were, however, a few instances where someone employing a slang phrase or colloquialism in a humorous or referential manner was mistakenly tagged as hateful because of a significant constraint while working on hate speech detection: the accurate classification of "hate."

Analytical Findings We performed an exploratory data analysis of our dataset described in Section 7 using Latent Dirichlet Allocation (LDA) (Jelodar et al., 2019) to understand the latent topics and derive semantic relationships of different themes and trends prevalent in our dataset. For example, a common variety of comments in our dataset shaming the inappropriate clothing style adopted by an influencer on the app led to many off-hand remarks that the model could not correctly identify, leading to a heavy reliance on LDA. This dependency of being familiar with specific topics in the dataset serves as a prospect that a hate speech tagger can incorporate will improve cases where a lack of context may lead to misclassification.

7 Analysing Bias in Comments

Hate comments against content creators that are defined as defamatory statements intended to portray

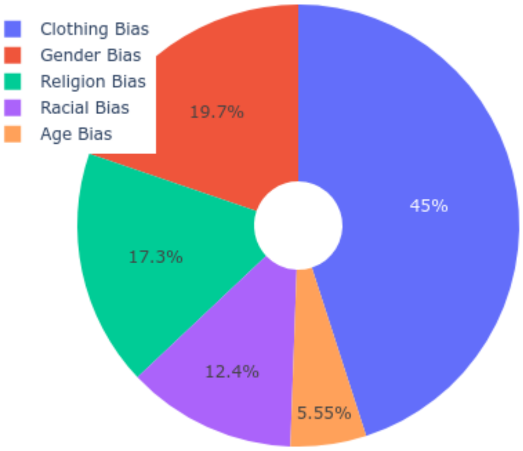

Figure 4: Bias-level percentage distribution of the 300 annotated samples

the artists are unfavorably within the broad category of offensive content on the Internet. A statement is discriminatory if it targets a person belonging to a particular social group segment for discriminatory reasons. For instance, targeting a specific gender, color, or religion might cause bias. To shed light on these biased attacks, we annotate 300 samples described in Table 2, and the distribution of these annotations is shown in Figure 4. For our sample, we randomly select comments labeled as Hate and which are in the Hindi language. While not exhaustive, the manually annotated labels offer a glimpse into the distribution, quality, and quantity of hateful comments. To further our analysis on a granular level, we perform Linear Discriminant Analysis (LDA) on the subsections of each data to identify the targeted sub-topics under the bias categories.

7.1 Gender Bias

Suppose gender bias is predominant in these social media platforms. In that case, it will perpetuate existing stereotypes, necessitating social media platforms to re-examine their algorithms as, ultimately, negatively shaping people's notion of a significant issue. Inferred from our annotation, we found a ratio of 11:1 tweets geared towards female content creators, which leads us to inspect further the subject on which the discrimination is based. For our analysis, we pick three topics to further our study on gender bias; solely gender, clothing, and age.

Clothing Bias Body covering or attire is an integral part of creating a persona that is available for perception by others. Clothing is also one of the most significant indications of gender identity, and being subject to a toxic environment, as portrayed by hate comments, can take a severe mental toll. It is vital to engage in a far more considerable effort to eradicate toxic attitudes learned consciously or unconsciously from mass media's modern 'schools.' Figure 4 shows that defamation against women's clothing is dominant compared to slander solely on gender or a woman's age. While labeling for clothing bias, we searched for clothing-specific keywords, ranging from clothing style to the variations in techniques used. For example, we observe clothing bias in comment; *"Kabhi kapde pahan Kar Bhi video banaa liya karo ladki ke naam per kalank Ho Tum" (Make a video after wearing clothes, you are a stigma in the name of the girl)*, by identifying the word "kapda"(clothes). We observed five major topics prevalent in our dataset along with their top keywords, as shown in Table 3, obtained from topic modeling using LDA.

Table 3: Clothing-level topics obtained from Latent Discriminatory Analysis with their top words

Topic	**Top Words**
Harassing the influencer based on clothing choice	Kapde (clothes), Pehn (wear), Tarika (style), N*ngi (naked)
Suggesting to change clothing style	video, full, kapde (clothes), pehno (wear)
Implying the influencer wants more followers	Followers, body, like, chahiye (want)
Implying that the content isn't Family friendly	Family, kapde (clothes), utaar (remove), problem
This topic infers the support of others against the negative comment section	comment, gande (bad), apko (you), karte (do)

Solely based on gender To differentiate comments based on their target gender demographic, we analyzed the "gender" attributed to important words. For example, in the comment; *"Saal** dikhawa karti hai suwar" (sister-in-law pretends to be a pig)*, certain words like "karta" (male) and "Karti" (female) both mean "do" / "be" but are gender specific along with certain keywords which are used to refer to women in a derogatory manner (for example *"saal**"* below means sister-in-law via a direct translation, however, it used in a negative connotation).

Age Bias While labeling comments based on age, we searched for specific keywords that negatively refer to someone's age. For example, in analysis; *"Are aunty apne beti ke kpde phn liya kya" (Hey aunty have you put on your daughter's clothes?)*, "aunty" is used to refer to elderly women.

Still, it can be used negatively depending on the words used alongside it. Despite the limited age bias in our sample size, we can't ignore the fact that some comments target a content creator's age. We are further cementing the gender bias discussed above. The top words obtained from the sample dataset after performing topic modeling were: Aunty(aunt), Maa(mother), Chudail(witch), and Bhuda(old), which in itself shows three out of four words directly targeting elder female content creators.

7.2 Racial Bias

In order to determine racial bias in the comments, we searched for keywords referring to the color of an individual's skin (for example, kala/kali- is used to address someone with a darker skin tone, and similarly, gora/gori is used to address someone with a fair skin tone). Topic modeling was used to obtain the following top words: kali(black), gori(white), chipkali(lizard), kalank(tainted). Based on the keyword search, most of the hateful comments associated with a racial bias had a close correlation with an individual's attire (clothing bias) or the public reception towards their body. In an example comment; *"Pari nahi tu kali chudail h apne mann me hi pari banti firti" (You are not a fairy, you're a black witch. You're only a fairy in your dreams)*, the word "kali (black)" is used in a negative light; attacking someone on racial grounds.

7.3 Religion Bias

Based on the results obtained from topic modeling, we could see a strong correlation between religious bias, clothing bias, and female bias, as many comments undermined women based on their religious affiliation and their choice of clothing. The top words observed were: Mulla(Muslim), Hindu(Hinduism), Sardar(Sikh), and Islam(Muslim). An Example:*"Aap musalman hokar bhi aise kapde pahnati ho kuchh to sharm karo adla pakshi" (Even though you are a Muslim, you wear such clothes, you should be ashamed.)*

8 Conclusion & Future Work

Any negative statement based on identification (such as gender, caste, or religion) rather than comments supporting the formation of an inclusive community should be avoided. To achieve this goal, we test various deep learning approaches on the Moj Multilingual Abusive Comment Identification dataset having thirteen distinct regional languages and constructed a model that outperforms our baselines. To advance our research, we manually annotate 300 data points with bias labels (gender, clothing, age, religion, race). A common thread that ties together other hateful comments and biases observed is the reference to clothes and how people perceive clothes as being inappropriate. Suppose we follow this through-line of hate geared towards the choice of clothing. In that case, the fact that most hateful comments are targeted towards female influencers hearkens to a societal issue of objectification and dehumanization that makes women prone to attacks and libel.

The model architecture can be improved in the future by testing other feature selection methods, elevating its overall performance while working with code-mixed languages. Second, research has shown the importance of context for hate speech classification. Certain cases arise where there is a lack of contextual information, causing our best model to misclassify specific entries where even humans would struggle. This may be mitigated by developing a more robust pipeline by incorporating steps such as co-reference analysis and sarcasm detection. Third, to comprehend where the model fails, there is a need for a detailed investigation of false positives and negatives. Furthermore, the research may be further carried out to analyze bias on other social media platforms.

9 Ethical Statement

Using datasets and algorithms for hate speech detection can have beneficial and harmful effects. We want to be clear that our intention is not to advance any discourse (biased or otherwise). Instead, by providing a more balanced real-world view of the discussion against content creators in India, we hope to educate the audience about the distorted commentary perspectives in India. Through research and analysis in this area, we hope to create more reliable platforms for discussing creators on social media.

References

Luis Aguirre and Emese Domahidi. 2021. Problematic content in spanish language comments in youtube videos about venezuelan refugees and migrants. *Journal of Quantitative Description: Digital Media*, 1.

Massih R Amini, Cyril Goutte, and Nicolas Usunier. 2010. Combining coregularization and consensus-based self-training for multilingual text categorization. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 475–482.

Mohamed Amine Bentaallah and Mimoun Malki. 2012. The use of wordnets for multilingual text categorization: A comparative study. In *ICWIT*, pages 121–128.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Ona De Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nicola Döring and M. Rohangis Mohseni. 2020. Gendered hate speech in youtube and younow comments: Results of two content analyses. *Studies in Communication and Media*.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Mai Ibrahim, Marwan Torki, and Nagwa El-Makky. 2018. Imbalanced toxic comments classification using data augmentation and deep learning. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 875–878. IEEE.

Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. 2019. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)*, pages 1–11.

Ximing Li, Changchun Li, Jinjin Chi, Jihong Ouyang, and Chenliang Li. 2018. Dataless text classification: A topic modeling approach with document manifold. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 973–982.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th forum for information retrieval evaluation*, pages 14–17.

S Mittal and P Dhyani. 2015. Multilingual text classification. *Int. J. Eng. Res. Technol.(IJERT)*, 4(3).

Bhagirath P Prajapati, Sanjay Garg, and Mahesh H Panchal. 2009. Automated text categorization with machine learning and its application in multilingual text categorization. In *National Conference on Advance Computing-NCAC09, Vallabh Vidyanagar, Anand, Gujarat, India*, pages 204–209.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2017. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*.

Sayar Ghosh Roy, Ujwal Narayan, Tathagata Raha, Zubair Abid, and Vasudeva Varma. 2021. Leveraging multilingual transformers for hate speech detection. *arXiv preprint arXiv:2101.03207*.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.

A Appendix

A.1 Dataset

Moj[2] is India's largest short-video app for multiple regional languages. We chose to use Moj as the basis of our study primarily because it facilitates the use of numerous regional languages and, in doing so, captures the sentiments of different communities at a granular level which cant be achieved with other social media platforms. Another factor that we weighed heavily is that because Moj was recently introduced, its hate flagging capabilities are not as well developed as the internationally established social media platforms, where content moderation removes extremely hateful comments and doesn't accurately depict how a community can spread online hate.

The Moj Multilingual Abusive Comment dataset provided by IIIT-D had the following characteristics:

- The human-annotated dataset was split into two sections: training and testing, each with 665k and 74k samples, respectively.
- The distribution of Abusive and Not Abusive samples was 312k and 352k, respectively.
- All the comments in the dataset are annotated according to the language used. There are instances where similar words in Hindi are code-mixed to create two variants based on the script used, namely Devanagari and Roman-Hindi. Similarly, regional languages like Marathi, Haryanvi, and Rajasthani are variants of the Devanagiri script, were code mixed with English, along with the other regional languages that follow their script (Kannada, Malayalam and Odia-Brahmi, Bengali-Bangla, Bhojpuri-Kathi, Tamil, Telugu-Abugida script a variant of Brahmi Script).
- The test dataset used in this research was not disclosed to the competitors and is not publically available as it was a part of the Moj Multilingual Abusive Comment Identification Challenge hosted by IIIT-D [3].

A.2 Training Strategy

- We noticed a modest difference between GPU and TPU accelerators: models trained on GPU perform significantly better. However, because the experimental time on TPU was shorter, we decided to use it for most of our trials which can be seen in Table 4.

Table 4: Time taken for each Epoch in hours

Info	XLM-Roberta
Accelerator	Tesla P100
Time Taken (hr)	11.54
EPOCH 1	3.86
EPOCH 2	3.72
EPOCH 3	3.49

- We also tried truncation sizes of 64, 128, and 256 and settled on 128 for the input text.
- We used different alpha, beta, and gamma values based on the test f1 score of each model to assign a higher weightage to the model that performed better. On conducting an exhaustive analysis of different combinations of alpha, beta, and gamma, we concluded that our model performed the best for the values of 0.35,0.33,0.34.
- Post-processing: Based on our findings, raising the threshold offered us an advantage. Thus we chose to adjust the thresholds for each language. After experimenting with various thresholds, we discovered that the numbers in Table 5 produced the best results.

[2] https://apps.apple.com/in/app/moj-short-video-app/id1523457550

[3] https://www.kaggle.com/competitions/iiitd-abuse-detection-challenge

Table 5: Language Wise Inference Threshold

Language	**threshold**
Marathi	0.56
Malayalam	0.52
Hindi	0.58
Telugu	0.62
Tamil	0.51
Odia	0.4
Gujarati	0.5
Bhojpuri	0.52
Haryanvi	0.6
Assamese	0.55
Kannada	0.5
Rajasthani	0.5
Bengali	0.55

- Psuedo Labelling: We continued the process of training our model over the course of two iterations, reintegrating examples from the test dataset that provided a prediction probability of greater than 90% to our training dataset giving us a boost of 1% in the test F1 score.

A.3 Annotation Guidelines

Due to the Hindi language's highest density and annotator proficiency, we employed stratified sampling to create a sample size of 300 randomly selected comments in the language. We examined the effects of various biases on the classification of hate using this sample as our starting point. Due to the dataset's limited annotation for hate categorization in the competition, we had three undergraduate students annotate each comment in accordance with the following guidelines: like a comment would be considered biased based on religion if it contained words relating to identifying a person based on their religion like; "Islam", "Islaam","Molla", "Mulla", "Muslim", "Musalman", "Isai", "Christ", "Singh", "Sardar", etc. Similarly, the details for the other categories are provided while analysis in Section 7.