StereoHate: Towards identifying Stereotypical Bias and Target group in Hate Speech Detection

Krishanu Maity\textsuperscript{1}, Nilabja Ghosh\textsuperscript{2}, Raghav Jain\textsuperscript{1}, Sriparna Saha\textsuperscript{1} and Pushpak Bhattacharyya\textsuperscript{3}

\textsuperscript{1} Department of CSE, Indian Institute of Technology Patna, India, \textsuperscript{2} Department of Computer Science, Ramakrishna Mission Vivekananda Educational and Research Institute, India \textsuperscript{3} Department of CSE, Indian Institute of Technology Bombay, India

krishanu_2021cs19@iitp.ac.in

(Received xx xxx xxx; revised xx xxx xxx; accepted xx xxx xxx)

Abstract

Though Social media helps spread knowledge more effectively, it also stimulates the propagation of online abuse and harassment, including hate speech. It is crucial to prevent hate speech since it may have serious adverse effects on both society and individuals. Therefore, it is not only important for models to detect these speeches, but to also output explanations of why a given text is toxic. While plenty of research is going on to detect online hate speech in English, there is very little research on low-resource languages like Hindi and the explainability aspect of hate speech. Recent laws like the "right to explanations" of the General Data Protection Regulation have spurred research in developing interpretable models rather than only focusing on performance. Motivated by this, we create the first interpretable benchmark hate speech corpus HHES in the Hindi language, where each hate post has its stereotypical bias and target group category. Providing descriptions of internal stereotypical bias as an explanation of hate posts makes a hate speech detection model more trustworthy. Current work proposes a commonsense-aware unified generative framework, $C\text{Gen}Ex$ by reframing the multitask problem as a text-to-text generation task. The novelty of this framework is it can solve two different categories of tasks (generation and classification) simultaneously. We establish the efficacy of our proposed model ($C\text{Gen}Ex$-fuse) on various evaluation metrics over other baselines when applied to Hindi HHES dataset.

Disclaimer: The article contains profanility, an inevitable situation for the nature of the work involved. These in no way reflect the opinion of authors.

1. Introduction

The exponential increase in textual content due to the widespread use of social media platforms renders human moderation of such information untenable (Cao et al. 2020). Governments, media organizations, and researchers now view the prevalence of hate speech on online social media platforms as a major problem, particularly given how quickly it spreads and encourages harm to both individuals and society. Hate speech (Nockleby 1994) is any communication that intends to attack the dignity of a group based on characteristics such as race, gender, ethnicity, sexual orientation, nationality, religion, or other features. With the advancement of Natural Language Processing (NLP), numerous studies have suggested methods to detect hate speech automatically using traditional Machine Learning
Natural Language Engineering

(ML) (Dadvar et al. 2014; Dinakar et al. 2011; Reynolds et al. 2011) and deep learning approaches (Agrawal and Awekar 2018; Waseem and Hovy 2016; Badjatiya et al. 2017). However, it is crucial for artificial intelligence (AI) tools not only to identify hate speech automatically but also to generate the implicit bias that is present in the post in order to explain why it is hated. The advent of explainable artificial intelligence (AI) (Gunning et al. 2019) has necessitated the provision of explanations and interpretations for decisions made by machine learning algorithms. This requirement is crucial for establishing trust and confidence in the deployment of AI models. Additionally, recent legislation in Europe, such as the General Data Protection Regulation (GDPR) (Regulation 2016), has implemented a “right to explanation” law, further emphasizing the need for interpretable models. Consequently, there is a growing emphasis on the development of models that prioritize interpretability rather than solely focusing on improving performance through increased model complexity.

Stereotypical bias (SB) (Cuddy et al. 2009), a common unintentional bias, can be based on specific aspects such as skin tone, gender, ethnicity, demography, disability, Arab-Muslim origin, etc. Stereotyping is a cognitive bias that permeates all aspects of daily life and is firmly ingrained in human nature. Social stereotypes have a detrimental influence on people’s opinions of other groups and may play a crucial role in how people interpret words aimed towards minority social groups (Sap et al. 2019a). For example, earlier studies have demonstrated that toxicity detection models correlate texts with African-American English traits with more offensiveness than texts lacking such qualities (Davidson et al. 2019).

In the past decade, extensive research has been conducted to develop datasets and models for automatic detection of online hate speech in the English language (Agrawal and Awekar 2018; Waseem and Hovy 2016; Badjatiya et al. 2017). However, there is a noticeable scarcity of hate speech detection work in the Hindi language, despite its status as the fourth-most-spoken language globally, widely used in South Asia. Existing studies in this domain have primarily focused on enhancing the performance of hate speech detection using various models, often neglecting the crucial aspect of explainability. The emergence of explainable artificial intelligence (AI) has now necessitated the provision of explanations and interpretations for decisions made by machine learning algorithms, becoming a critical requirement in this field. For instance, debiasing techniques that incorporate knowledge of the toxic language may benefit from extra information provided by in-depth toxicity analyses in text (Ma et al. 2020). Furthermore, thorough descriptions of toxicity can make it easier for people to interact with toxicity detection systems (Rosenfeld and Richardson 2019).

To fill this research gap, in this work, we create a benchmark Hindi hate speech explanation dataset (HHES) that contains the stereotypical bias and target group category of a toxic post. To create HHES dataset, we manually translate the existing English Social Bias Frames (SBIC) (Sap et al. 2020a) dataset. Now, we have to develop an efficient multitask framework that can solve two different categories of tasks simultaneously, i.e., (i) sequence generation task (generate stereotypical bias as explanation) and (ii) classification task (identify the target group category).

Humans have the ability to learn multiple tasks simultaneously and apply the knowledge learned from one task to another task. To mimic this quality of human intelligence, researchers have been working on Multitask learning (MTL) (Caruana 1997) which is a training paradigm in which a model is trained with data from different closely related tasks in an attempt to efficiently learn the mapping and connection between these tasks. There have been many works that have shown that solving a closely related auxiliary task along with the main task increases the performance of the primary tasks (such as cyber-bullying detection (CD) (Maity and Saha 2021b), complaint identification Singh et al...
A typical multitask model consists of a shared encoder that contains representations from data of different tasks and several task-specific layers or heads attached to that encoder. However, there are many drawbacks of this approach such as negative transfer \cite{Crawshaw2020} (where multiple tasks instead of optimizing the learning process start to hurt the training process), model capacity \cite{Wu2019} (if the size of shared encoder becomes too large then there will be no transfer of information across different tasks) or optimization scheme \cite{Wu2019} (how to assign weights to different tasks during training). There are also several scalability issues with this approach of multitasking such as adding task-specific heads every time a new task has been introduced or changing the complete model architecture whenever a new combination of tasks has been introduced.

To overcome the challenges of MTL, we propose the use of a generative model to solve two different categories of tasks: classification (target group category) and generation (stereotypical bias). Rather than employing two separate models to address these tasks, we present a commonsense-aware unified generative multitask framework that can solve both tasks simultaneously in a text-to-text generation manner. We converted the classification task into a generation task, where the target output sentence is the concatenation of the classification task’s output tokens. In our proposed model, the input is text, such as a social media post, and the output is also text, representing the concatenation of stereotypes and target groups separated by a special character. For instance, given the input post “Bitches love Miley Cyrus and Rihanna because they speak to every girl’s inner ho,” the corresponding output or target sequence is ”<Women are sexually promiscuous> <Gender>.” In this example, “Women are sexually promiscuous” represents the stereotypical bias, and “Gender” is the target group category. As sentient beings, we use our common sense to establish connections between what is explicitly said and inferred. We employed Conceptnet to generate commonsense knowledge to capture and apply common patterns of real-world knowledge in order to draw conclusions or make decisions about a given post. For example, if the input sentence is “I was just pretending to be retarded!” Then some of the generated commonsense reasonings by Conceptnet are (i) “pretend requires imagination”, (ii) “retard is similar in meaning to an idiot”.

To sum up, our contributions are twofold:

1. HHES, a new benchmark dataset for explainable hate speech detection with target group category identification in the Hindi language has been developed.
2. To simultaneously solve two tasks, i.e., stereotypical bias/explanation (generation task) and identifying target group (classification task), a commonsense-aware unified generative framework (\textsc{CGenEx}) with reinforcement learning-based training has been proposed.

The organization of this article is as follows. A survey of all the previous works in this domain is explained in Section \ref{related_works}. Section \ref{dataset_creation} describes the process of dataset creation in detail. Section \ref{methodology} explains the proposed methodology and Section \ref{experiments} describes the experimental settings and results. This part also contains a detailed error analysis of our results.

\section{Related Works}

Hate speech is very reliant on linguistic subtlety. Researchers have recently provided a lot of attention to automatically identifying hate speech in social media. In this section, we will review recent works on detecting and explaining hate speech.

\footnote{The code and dataset will be made publicly available in the camera-ready version}
2.1 Hate speech Detection
Kamble et al. (Kamble and Joshi 2018) explored hate speech detection in code-mixed Hindi-English tweets. By employing three deep learning models with domain-specific embeddings, they achieved a significant improvement of 12% in F1-score compared to previous work that used statistical classifiers. The authors emphasized the ability of their models to capture the semantic and contextual aspects of hate speech, highlighting the value of domain-specific word embeddings. In this paper (Kumar et al. 2018), the authors address the increasing incidents of aggression and related behaviors on social media platforms by developing an aggression-annotated dataset of Hindi-English code-mixed data from Twitter and Facebook. The dataset, consisting of approximately 18k tweets and 21k Facebook comments, is annotated with a hierarchical tagset of aggression levels and types. This annotated dataset serves as a valuable resource for understanding and automatically identifying aggression, trolling, and cyberbullying on social media platforms. Maity et al. (Maity and Saha 2021a) introduces a benchmark corpus specifically designed for detecting cyberbullying targeted at children and women in the context of Hindi-English code-mixed language. By combining BERT, CNN, GRU, and Capsule networks, the authors develop a powerful model for classification, surpassing both conventional machine learning and deep neural network baselines. The model achieves an accuracy of 79.28% highlighting its effectiveness in identifying cyberbullying instances. By leveraging the power of the BERT language model, Paul and Saha (2020) developed a transformer-based method for identifying hate speech across multiple social media platforms. The approach involves fine-tuning BERT and implementing a straightforward classification model, leading to state-of-the-art performance on real-world datasets from Formspring, Twitter, and Wikipedia. Badjatiya et al. (2017) addresses the task of hate speech detection on Twitter by leveraging deep learning architectures and semantic word embeddings. Through extensive experimentation on a benchmark dataset of 16K annotated tweets, the study demonstrates that the proposed deep learning methods outperform state-of-the-art character n-gram and word TF-IDF methods by a significant margin of approximately 18% F1 points. The authors also highlight the superiority of certain combinations, such as LSTM with random embedding and GBDT, and provide evidence of the task-specific nature of the learned embeddings through word similarity comparisons. Watanabe et al. (2018) presents an approach for detecting hate speech on Twitter by leveraging patterns and unigrams collected from the training set as features for machine learning algorithms. The proposed method achieves high accuracies of 87.4% for binary classification (offensive vs. non-offensive) and 78.4% for ternary classification (hateful, offensive, or clean). The study highlights the importance of automatically identifying hate speech to filter out offensive content and proposes future work to expand the dictionary of hate speech patterns and analyze the presence of hate speech across different demographics. Davidson et al. (2017) addresses the challenge of distinguishing hate speech from other types of offensive language in social media. Using a crowd-sourced hate speech lexicon and a multi-class classification model, the study accurately categorizes tweets into hate speech, offensive language, or neutral content. The findings highlight the importance of precise classification, uncover insights into different types of hate speech, and emphasize the need to address social biases in hate speech detection algorithms.

2.2 Explainability/Bias
Zaidan et al. (Zaidan et al. 2007) proposed the concept of rationales, in which human annotators underlined a section of text that supported their tagging decision. Authors have examined that the usages of these rationales certainly improved sentiment classification
performance. (Mathew et al. 2020) introduce HateXplain, a comprehensive benchmark dataset that includes annotations from multiple perspectives, such as classification labels (hate, offensive, normal), target communities, and rationales based on which labeling decisions are made. The study evaluates state-of-the-art models on this dataset and highlights the limitations of high-performing classification models in terms of explainability. Furthermore, the findings demonstrate the importance of incorporating human rationales in training models to mitigate unintended bias and improve performance in hate speech detection.

Sridhar et al. (Sridhar and Yang 2022) developed the MixGEN model based on expert, explicit and implicit knowledge to explain toxic text by generating the stereotype of the post. They have experimented on SBIC social bias dataset collected from different social media like Twitter, Reddit, Gab, etc. The study highlights the strengths and weaknesses of different knowledge types and emphasizes the effectiveness of mixture and ensemble methods in leveraging diverse knowledge sources to generate high-quality text generations. To remove stereotypical bias in the hate speech detection task, authors in Badjatiya et al. (2019) propose a two-stage framework that includes heuristics to identify bias-sensitive words and novel strategies based on knowledge generalization for replacing these words. Experimental results using real-world datasets (WikiDetox and Twitter) demonstrate the effectiveness of the proposed methods in reducing bias without compromising overall model performance. The study highlights the potential of data correction techniques and provides qualitative analysis and examples to support the findings. (Karim et al. 2021) developed DeepHateExplainer, an explainable approach for hate speech detection in the under-resourced Bengali language. The authors preprocess Bengali texts and employ a neural ensemble method using transformer-based neural architectures to classify hate speech into political, personal, geopolitical, and religious categories. They utilize sensitivity analysis and layer-wise relevance propagation (LRP) to identify important terms and generate human-interpretable explanations. Evaluations against machine learning and neural network baselines demonstrate the superior performance of DeepHateExplainer. The study acknowledges potential limitations due to limited labeled data and proposes future directions for improvement and expansion.

2.3 Text Generation

Models such as GPT-2 (Radford et al. 2019) and GPT-3 are decoder-only transformer models that have been pre-trained on a large amount of text data that can generate fluent, coherent, and consistent text. Encoder-decoder Transformers consisting of BART (Lewis et al. 2020) and T5 (Raffel et al. 2020), have shown massive improvements and success in many NLP tasks such as summarization and translation. Recently, there are many attempts to use these generative models in solving non-generational tasks. (Yan et al. 2021) used the BART model to solve the task of aspect-based sentiment analysis. They proposed to convert all the aspect-based sentiment analysis tasks to a unified generation task. The BART model is implemented to generate the target sequence in an end-to-end process based on unified task generation. Similarly, (Wang et al. 2022) used the T5 model for solving named entity recognition as a generative problem. This enriches source sentences with task-specific instructions and answer options, then inferences from the entities and types in natural language. The T5 model is further trained for tasks such as entity extraction and entity typing.

After an in-depth literature review, we can conclude that most of the works on hate speech detection are in English, and there is no such work on the explainability of hate speech by generating the internal stereotypical bias in the Hindi language. In this work, we attempt to bridge this research gap.
3. Dataset Creation

This section discusses the developed benchmark Hindi hate speech explanation (stereotypes) dataset (HHES). To begin, we reviewed the literature for the existing hate speech datasets, which contain stereotypical bias and target groups. As per our knowledge, there is only one standard social bias dataset (SBIC) in English developed by (Sap et al. 2020a). The lack of any other publicly available dataset related to our work and the good structure of this dataset makes it the perfect choice for our purpose.

Technological advancements have revolutionized the way people express their opinions, particularly in low-resource languages. India, a country with a massive internet user base of 1010 million, exhibits significant linguistic diversity. Among the numerous languages spoken in India, Hindi holds a prominent position as one of the official languages, with over 691 million speakers. Consequently, a substantial portion of text conversations on social media platforms in India occurs in the Hindi language. This phenomenon highlights the significance of Hindi as the primary medium of communication for the majority of users in the country.

We have manually annotated the existing English SBIC dataset to create the Hindi hate speech explanation dataset (HHES). The annotation process was overseen by two proficient professors who have extensive expertise in hate speech and offensive content detection. The execution of the annotation task was carried out by a group of ten undergraduate students who were proficient in both Hindi and English. These students were recruited voluntarily through the department email list and were provided compensation in the form of gift vouchers and an honorarium for their participation. To ensure consistency and accuracy in the translation process, we initiated the annotation training phase with a set of gold-standard translated samples. Our expert annotators randomly selected 300 samples and manually translated them from English to Hindi. Through collaborative discussions, any differences or discrepancies in the translations were resolved, resulting in the creation of 300 gold-standard manually annotated samples encompassing toxic posts and their corresponding stereotypes. To facilitate the training of novice annotators, these annotated examples were divided into three sets, each containing 100 samples. This division allowed for a three-phase training procedure in which novice annotators received guidance and feedback from the expert annotators. After the completion of each training phase, the expert annotators collaborated with the novice annotators to rectify any incorrect annotations and provide further guidance. Upon the conclusion of the three-phase training process, the top ten annotators were selected based on their performance. These annotators were chosen to annotate the entire dataset, and the workload was evenly divided among them. Therefore, each post was translated by one of the selected annotators.

However, we acknowledge that despite our diligent efforts, there may be cases where the translation does not precisely replicate the original post due to the inherent difficulties of cross-lingual translation and the complexities of social media language.

The number of training, validation, and test samples in HHES dataset are 12110, 1806, and 1924, respectively. The detailed distribution of target group category classes is shown in Table 1.

Further, we have engaged three senior annotators (Master’s students in Linguistics) to verify the translation quality in terms of fluency (F) and adequacy (A) as mentioned in (Ghosh et al. 2022). Fluency evaluates whether the translation is syntactically correct or not, whereas Adequacy checks the semantic quality. Each annotator marked every
Table 1: Train, validation, and test split distribution of Target group category in HHES dataset.

<table>
<thead>
<tr>
<th>Split</th>
<th>Race</th>
<th>Gender</th>
<th>Social</th>
<th>Body</th>
<th>Culture</th>
<th>Disabled</th>
<th>Victim</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>3462</td>
<td>3294</td>
<td>538</td>
<td>369</td>
<td>2134</td>
<td>713</td>
<td>1600</td>
<td>12110</td>
</tr>
<tr>
<td>Val</td>
<td>681</td>
<td>441</td>
<td>72</td>
<td>54</td>
<td>342</td>
<td>92</td>
<td>124</td>
<td>1806</td>
</tr>
<tr>
<td>Test</td>
<td>716</td>
<td>439</td>
<td>75</td>
<td>47</td>
<td>421</td>
<td>102</td>
<td>124</td>
<td>1924</td>
</tr>
</tbody>
</table>

Table 2: Two examples from HHES dataset

<table>
<thead>
<tr>
<th>Post</th>
<th>Target Stereotype</th>
<th>Target group Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hindi: मैं आज तुम्हारे लिए नहीं गया हूँ. मुझे वापस अपने निकला हाथ दे।</td>
<td>हिंदी के लिए हैं।</td>
<td>Race</td>
</tr>
<tr>
<td>English: What up fake ass hoe? I’m not gone fought you today. Gimme my nigga hand back*</td>
<td>nigga is a slang term for black people</td>
<td></td>
</tr>
<tr>
<td>Hindi: और मैं एक कूड़ा-कूड़ा बदचलन के बारे में जानी नहीं जा सकती।</td>
<td>औरतें बदचलन हैं।</td>
<td>Gender</td>
</tr>
<tr>
<td>English: And I can’t be known for fucking wit a trash bitch</td>
<td>women are bitches</td>
<td></td>
</tr>
</tbody>
</table>

*Fluency - 5: Flawless, 4: Good, 3: Non-native, 2: Disfluent, 1: Incomprehensible; Adequacy - 5: All, 4: Most, 3: Much, 2: Little, 1: None

translated sentence with an ordinal value from a scale of 1-5 for both F and A. We attain high average F and A scores of 4.23 and 4.58, respectively, illustrating that the translations are of good quality. In Table 2, some examples of the HHES dataset are shown.

4. Methodology

In this work, we have proposed CGenEx (shown in Figure 1), a commonsense-aware unified generative framework for generating stereotypical bias to explain why an input post is hateful and identify the target group category. Detailed descriptions of the proposed models are described below.

4.1 Commonsense-aware Generative Framework (CGenEx)

We propose a text-to-text generation paradigm for solving hate speech explanations and identifying target group categories in a unified manner. To transform this problem into a text generation problem, we first construct a natural language target sequence, $Y_i$, for input sentence, $X_i$, for training purposes by concatenating the explanations (stereotypical bias) and target group. Finally, the target sequence $Y_i$ is represented as:

$$Y_i = \{<St><Tg>\}$$ (1)

where St and Tg represent the corresponding stereotypical bias and target group of an input post $X_i$, respectively.

We have added special characters after each task’s prediction as shown in Equation 1 so that we can extract task-specific predictions during testing or inference. Now, both the input sentence and the target are in the form of natural language to leverage large pre-trained sequence-to-sequence models for solving this task of text-to-text generation.

Now the problem can be reformulated as: given an input sequence $X$, the task is to generate an output sequence, $Y'$, containing all the predictions defined in Equation 1.
using a generative model defined in Equation 2:

\[ Y' = G(X) \]  (2)

where \( G \) is a generation model. We divide our approach into three steps: 1) Commonsense extraction module, 2) Commonsense aware transformer model and 3) Reinforcement learning-based training.
4.1.1 Sequence to Sequence Learning (Seq2Seq):
This problem of a text-to-text generation defined in equation \( 2 \) can easily be solved with the help of a Sequence to Sequence model which consists of two modules: 1) Encoder and 2) Decoder. We employed the pre-trained BART (Lewis et al. 2020) and T5 (Raffel et al. 2020) models as the Sequence to Sequence models in our proposed model (CGenEx).

**BART**: BART is an encoder-decoder-based transformer model which is mainly pre-trained for text generation tasks such as summarization and translation. BART is pre-trained with various denoising pretraining objectives such as token masking, sentence permutation, sentence rotation etc.

**T5**: T5 is also an encoder-decoder-based transformer model which aims to solve all the text-to-text generation problems. The main difference between BART and T5 is the pre-training objective. In T5, the transformer is pre-trained with a denoising objective where 15% of the input tokens are randomly masked and the decoder tries to predict all these masked tokens whereas, during pre-training of BART, the decoder generates the complete input sequence.

4.1.2 Commonsense Extraction Module
Commonsense reasoning in NLP models is the ability of the model to capture and apply common patterns of real-world knowledge in order to draw conclusions or make decisions about a particular text or data set (Sap et al. 2020b). This type of reasoning allows the model to draw inferences. Incorporating commonsense reasoning in language models can help to more accurately capture the underlying intentions and context behind the speech. We employ a commonsense extraction module to provide more context in the form of commonsense reasoning to the input text so that model can incorporate knowledge regarding social entities and events involved in the input text. We use ConceptNet (Speer et al. 2017) as our knowledge base for the commonsense extraction module. At first, we feed the input text, \( X_i \), to the Commonsense Extraction Module to extract the top 5 commonsense reasoning triplets using the same strategy as mentioned in (Sridhar and Yang 2022) where a triplet consists of two entities and a connection/relation between these two entities which is then converted into a single sentence. Formally, to get the top 5 triplets from Conceptnet, we take the nouns, verbs, and adjectives from the input and search for related triplets in ConceptNet. Then, we sort them in order of the combination of their IDF score and the edge weight of the triplets and then will select the top 5 triplets. To obtain the final commonsense reasoning \( CS \) for each input text, \( X_i \), we concatenate these five commonsense reasonings together.

4.1.3 Commonsense Aware Transformer
To leverage the commonsense reasoning \( CS \) obtained from the Commonsense Extraction Module, we have proposed two variations of commonsense aware encoder-decoder architecture (CGenEx-con and CGenEx-fuse) that are capable of incorporating \( CS \) in their sequence-to-sequence learning process. We employed the pre-trained BART (Lewis et al. 2020) and T5 (Raffel et al. 2020) models as the base sequence to sequence models.

4.1.4 CGenEx-con (Concatenation based CGenEx)
Given an input text \( X_i \) and corresponding commonsense reasoning \( CS \), the task to generate the target sequence, \( Y'_i \), can be modeled as the following conditional text generation model: \( P_\theta(Y'_i|X_i, CS) \), where \( \theta \) is a set of model parameters. CWHSI-Con models this conditional probability as follows:
We first concatenate the tokens of the input text, $X_i$, and the commonsense reasoning, $CS$ to provide us with a final input sequence as follows: $T_i = X_i \oplus CS$. Now, given a pair of input sentences and target sequence $(T_i, Y_i)$, the first step is to feed $T_i$ to the encoder module to obtain the hidden representation of input defined as

$$H_{EN} = G_{Encoder}(T_i) \quad (3)$$

where $G_{Encoder}$ represents encoder computation.

After obtaining the hidden representation, $H_{EN}$, we will feed $H_{EN}$ and all the output tokens till time step $t - 1$ represented as $Y_{<t}$ to the decoder module to obtain the hidden state at time step $t$ as defined in below equation.

$$H'_{DEC} = G_{Decoder}(H_{EN}, Y_{<t}) \quad (4)$$

where $G_{Decoder}$ denotes the decoder computations.

The conditional probability for the predicted output token at $t^{th}$ time step, given the input and previous $t - 1$ predicted tokens is calculated by applying the softmax function over the hidden state, $H'_{DEC}$, as follows:

$$P(Y'_t | X, Y_{<t}) = F_{softmax}(H'_{DEC}W_{Gen}) \quad (5)$$

where $F_{softmax}$ represents softmax computation and $W_{Gen}$ denotes weights of our model.

### 4.1.5 CGenEx-Fuse (Fusion based CGenEx)

To fuse the information from both commonsense and input text, we have proposed a commonsense aware encoder (shown in Figure 1), an extension of the original transformer encoder Vaswani et al. (2017). At first, the input text, $X_i$, is tokenized and converted into a sequence of embeddings. Then positional encodings are added to these token embeddings to retain their positional information before feeding input to the proposed commonsense aware encoder. Our commonsense aware encoder is composed of three sub-layers: 1) Multi-head Self-Attention (MSA), 2) Feedforward Network (FFN) and 3) Commonsense Fusion (CSF). MSA and FFN are standard sub-layers as used in the original transformer encoder Vaswani et al. (2017). We have added a CSF sub-layer as a means to fuse the commonsense knowledge in our model which works as follows:

After obtaining the encoded representation, $H_{EN}$ from the first two sub-layers (MSA and FFN), we feed this $H_{EN}$ and commonsense feature vector $G_{CS}$ to the CSF sub-layer. Unlike the standard transformer encoder where we project the same input as query, key, and value, in CWHSI-Fuse, we implement a context-aware self-attention mechanism inside CSF to facilitate the exchange of information between $H_{EN}$ and $G_{CS}$, motivated by Yang et al. (2019). We create two triplets of queries, keys, and values matrices corresponding to $H_{EN}$ and $G_{CS}$, respectively: $(Q_x, K_x, V_x)$ and $(Q_{cs}, K_{cs}, V_{cs})$. Triplets $(Q_x, K_x, V_x)$ are generated by linearly projecting the input text representation, $H_{EN}$, whereas triplets $(Q_{cs}, K_{cs}, V_{cs})$ are obtained through gating mechanism as given in Yang et al. (2019) which works as follows:

To maintain a balance between fusing information from commonsense representation, $G_{CS}$, and retain original information from text representation, $H_{EN}$, we learn matrices $\lambda_K$ and $\lambda_V$ to create context-aware $K_{cs}$ and $V_{cs}$ (Equation 6).

$$\begin{bmatrix} K_{cs} \\ V_{cs} \end{bmatrix} = (1 - \begin{bmatrix} \lambda_K \\ \lambda_V \end{bmatrix}) \begin{bmatrix} K_x \\ V_x \end{bmatrix} + \begin{bmatrix} \lambda_K \\ \lambda_V \end{bmatrix} (G_{CS} \begin{bmatrix} U_K \\ U_V \end{bmatrix}) \quad (6)$$
where $U_K$ and $U_V$ are learnable parameters and matrices $\lambda_K$ and $\lambda_V$ are computed as follows:

$$
\begin{bmatrix}
\lambda_K \\
\lambda_V
\end{bmatrix} = \sigma
\left(
\begin{bmatrix}
K_x & W_K^X \\
V_x & W_V^X
\end{bmatrix} + G_{CS}
\begin{bmatrix}
U_K & W_K^{CS} \\
U_V & W_V^{CS}
\end{bmatrix}
\right)
$$

(7)

where $W_K^X$, $W_V^X$, $W_K^{CS}$ and $W_V^{CS}$ all are learnable parameters. $\sigma$ represents the sigmoid function computation.

After obtaining $K_{cs}$ and $V_{cs}$, we apply the dot product attention based fusion method over $Q_x$, $K_{cs}$ and $V_{cs}$ to obtain the final commonsense aware input representation, $Z$, computed as:

$$
Z = \text{softmax} \left( \frac{Q_x K_{cs}^T}{\sqrt{d_k}} \right) V_{cs}
$$

(8)

At last, we feed this commonsense aware input representation vector, $Z$, to an autoregressive decoder following the same decoder computations defined in equation 4.

**4.1.6 Reinforcement Learning based Training**

We initialize our model’s weights $\theta$ with weights of a pre-trained sequence to sequence generative model. We then fine-tune the model with the following two training objective functions: 1) Negative log-likelihood, i.e., the maximum likelihood estimation (MLE) objective function, which works in a supervised manner to optimize the weights, $\theta$, as defined in Equation 9.

$$
\max_{\theta} \prod_{t=0}^{T} P_{\theta}(Y'_t|X_i, Y_{<t})
$$

(9)

2) On top of the maximum likelihood estimation (MLE) objective function, we also employ a reward-based training objective function. Inspired from (Sancheti et al. 2020), we use a BLEU (Papineni et al. 2002a) based reward function. We define BLEU based Reward $R_{BLEU}$ in Equation 10.

$$
R_{BLEU} = (\text{BLEU}(Y'_i, Y_i) - \text{BLEU}(Y_i^g, Y_i)),
$$

(10)

where $Y'_i$ denotes the output sequence sampled from the conditional probability distribution at each decoding time stamp and $Y_i^g$ denotes the output sequence obtained by greedily maximizing the conditional probability distribution at each time step. To maximize the expected reward, $R_{BLEU}$ of $Y'_i$, we use the policy gradient technique which is defined in Equation 11.

$$
\nabla_{\theta} J(\theta) = R_{BLEU} \cdot \nabla_{\theta} \log P(Y'_i|X_i, CS; \theta)
$$

(11)

**4.1.7 Inference**

During the training process, we have access to both the input sentence ($X_i$) and target sequence ($Y_i$). Thus, we train the model using the teacher forcing approach, i.e., using the target sequence as the input instead of tokens predicted at prior time steps during the decoding process. However, the inference must be done in an autoregressive manner as we don’t have the access to target sequences to guide the decoding process. After obtaining the predicted sequence $Y_i'$, we split that sequence around the special character ($<>$) to get the corresponding predictions for different tasks; stereotypical bias, and target group as described in Equation 1.
5. Experiments and Results

This section contains a detailed explanation of the experimental settings and the corresponding results. Certain standard baseline models are also mentioned for evaluating our results. The final part of this section is the ablation study and error analysis.

5.1 Experimental Settings

In this section, we detail various hyperparameters and experimental settings used in our work. We have performed all the experiments on Tyrone machine with Intel’s Xeon W-2155 Processor having 196 Gb DDR4 RAM and 11 Gb Nvidia 1080Ti GPU. We have executed all of the models five times, and the average results have been reported. We have used mBART and mT5 as the base model for both GenEx-Con and GenEx-Fuse. Both these models are trained for a maximum of 110000 epochs and a batch size of 16. Adam optimizer is used to train the model with an epsilon value of 0.00000001. All the models are implemented using Scikit-Learn\footnote{https://scikit-learn.org/stable/} and pytorch\footnote{https://pytorch.org/} as a backend.

5.2 Standard Baselines

We have developed the following standard baselines for a fair comparison with our proposed model.

Classification Baselines

We have experimented with four standard baselines as proposed in \cite{Mathew2020} for the target group identification task. BERT \cite{Devlin2018} is a language model based on a bidirectional transformer encoder with a multi-head self-attention mechanism. We selected mBERT, which has been trained in 104 different languages, including Hindi. mBERT generated sequence output has been considered as input embedding to the first three baselines.

(1) **CNN-GRU**: The sequence output from BERT, with dimensions $128 \times 768$, is passed through 1D CNN layers. These layers consist of three kernel sizes ($1, 2, 3$) and 100 filters for each size. The resulting convoluted features are then fed into a GRU layer. The hidden output from the GRU layer is passed to a Fully Connected (FC) layer with 100 neurons, followed by an output softmax layer.

(i) **BLEU**: One of the earliest metrics to be used to measure the similarity between two phrases is BLEU. It was first proposed for machine translation and is described as the geometric mean of n-gram precision scores times a brevity penalty for short sentences. We apply the smoothed BLEU in our experiments as defined in \cite{Lin2004}.

(ii) **ROUGE-L**: ROUGE was first presented for the assessment of summarization systems, and this evaluation is carried out by comparing overlapping n-grams, word sequences, and word pairs. In this work, we employ the ROUGE-L version, which measures the longest common subsequences between a pair of phrases.

(iii) **BERTScore**: It is a similarity metric for text generation tasks based on pre-trained BERT contextual embeddings. BERTScore uses a weighted aggregate of cosine similarities between two phrases’ tokens to determine how similar they are.
(2) **BiRNN**: The input is fed into a Bidirectional GRU (Bi-GRU) with 128 hidden units, generating a 256-dimensional hidden vector. This hidden vector is then passed to an FC layer, followed by output layers for the final class prediction.

(3) **BiRNN-Attention**: Similar to the previous baseline model, but with the addition of an attention layer between the Bi-GRU and FC layers.

(4) **BERT-finetune**: In this approach, the mBERT model is fine-tuned by adding an output softmax layer on top of the "CLS" output.

**Generation Baselines**: We use mBART [Liu et al. (2020)] and T5 [Raffel et al. (2020)] as the baseline text-to-text generation models. We fine-tune these models on the proposed dataset with the training objective defined in Equation 9. In a single task setting, the output sequence is either the stereotype or target group category, depending on which task you want to solve. In the case of multitasking, the output sequence is the concatenation of the stereotype and target group category.

Table 3.: Results of different baselines and the two proposed frameworks, CGenEx-Con and CGenEx-Fuse in a multi task setting. For the target tasks the results are in terms of macro-F1 score (F1), Accuracy (Acc) and Matthews correlation coefficient (MCC) values. F1, Acc and MCC metrics are given in %. The maximum scores attained are represented by bold-faced values; Gray Highlight represents statistically significant results;

<table>
<thead>
<tr>
<th>Model</th>
<th>Stereotype</th>
<th>Target</th>
<th>Accuracy</th>
<th>F1-Score</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>ROUGE-L</td>
<td>BERTScore</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Standard Baselines</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNN-GRU</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>60.23</td>
<td>43.33</td>
</tr>
<tr>
<td>BiRNN</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>60.81</td>
<td>43.99</td>
</tr>
<tr>
<td>BiRNN+Attention</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>62.33</td>
<td>44.31</td>
</tr>
<tr>
<td>BERT-Finetune</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>65.41</td>
<td>47.37</td>
</tr>
<tr>
<td>mT5-ST</td>
<td>36.38</td>
<td>39.87</td>
<td>78.52</td>
<td>62.43</td>
<td>45.78</td>
</tr>
<tr>
<td>mT5-MT</td>
<td>37.58</td>
<td>41.14</td>
<td>79.72</td>
<td>64.12</td>
<td>46.88</td>
</tr>
<tr>
<td>mBART-ST</td>
<td>41.47</td>
<td>45.72</td>
<td>81.12</td>
<td>81.23</td>
<td>67.35</td>
</tr>
<tr>
<td>mBART-MT</td>
<td>42.25</td>
<td>46.33</td>
<td>82.28</td>
<td>83.12</td>
<td>72.44</td>
</tr>
<tr>
<td><strong>Proposed model (CGenEx) Single task</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mT5:CGenEx-con</td>
<td>37.14</td>
<td>40.89</td>
<td>79.12</td>
<td>65.53</td>
<td>47.84</td>
</tr>
<tr>
<td>mT5:CGenEx-fuse</td>
<td>38.62</td>
<td>42.37</td>
<td>80.46</td>
<td>66.92</td>
<td>48.07</td>
</tr>
<tr>
<td>mBART:CGenEx-con</td>
<td>41.87</td>
<td>45.93</td>
<td>82.77</td>
<td>83.54</td>
<td>72.66</td>
</tr>
<tr>
<td>mBART:CGenEx-fuse</td>
<td>42.95</td>
<td>46.74</td>
<td>83.09</td>
<td>83.83</td>
<td>72.78</td>
</tr>
<tr>
<td><strong>Proposed model (CGenEx) Multi task</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mT5:CGenEx-con</td>
<td>38.25</td>
<td>42.14</td>
<td>80.75</td>
<td>67.96</td>
<td>48.67</td>
</tr>
<tr>
<td>mT5:CGenEx-fuse</td>
<td>38.88</td>
<td>42.97</td>
<td>81.63</td>
<td>68.12</td>
<td>48.86</td>
</tr>
<tr>
<td>mBART:CGenEx-con</td>
<td>43.12</td>
<td>47.08</td>
<td>83.89</td>
<td>84.36</td>
<td>72.98</td>
</tr>
<tr>
<td>mBART:CGenEx-fuse</td>
<td><strong>44.23</strong></td>
<td><strong>48.83</strong></td>
<td><strong>85.27</strong></td>
<td><strong>84.77</strong></td>
<td><strong>73.24</strong></td>
</tr>
</tbody>
</table>
5.3 Findings from Experiments

Table 3 shows and compares the results of stereotypical bias generation (SBG) and target group category identification (TI) tasks of our proposed model, CGenEx with different baseline models in both single tasks (one task at a time) and multitask settings. From all these reported results, we can conclude the following:

(1) It can be observed from Table 3 that BERT-finetune performs better in TI task as compared to other standard baselines (CNN-GRU, BiRNN, BiRNN+Attention). However, all the generative baselines based on mBART and our proposed models (CGenEx-Con and CGenEx-Fuse) can outperform the BERT-finetune model by a huge margin showing the superiority of pre-trained sequence-to-sequence language models.

(2) When we compare the performance of generative baselines, mBART always performs better than mT5 in both single-task (ST) and multitask (MT) settings. Like, mBART-ST outperforms the mT5-ST model by a margin of 1) 18.80% and 21.57% in accuracy and F1 score for TI task, respectively, and 2) 5.09%, 5.85% and 2.60% in BLEU, ROUGE-L and BERTScore metrics for SBG task, respectively. Similar trends are also observed for proposed models, i.e., any variants of our proposed model (CGenEx-con or CGenEx-fuse), when embedded with mBART, perform better compared to one embedded with mT5. This finding established that mBART is significantly better at handling Hindi data than mT5.

(3) It is also evident from Table 3 that our proposed model CGenEx-Fuse always outperforms the baselines by a significant margin for both tasks. mBART:CGenEx-fuse outperforms the best generative baselines, mBART-MT, with an improvement of (i) 1.98% and 2.09% in BLEU, and ROUGE-L metrics for the SBG task, respectively. and (ii) 0.94% in F1 score for TI task. But another variant of our proposed model CGenEx-con slightly underperforms the best baseline mBART-MT in single task settings (mBART-ST: 41.87, 45.93; ST-mBART:CGenEx-con: 41.47, 45.72) and almost comparable results (mBART-MT: 72.25, 46.33; MT-mBART:CGenEx-con: 42.32, 46.38) in multitasking settings in terms of BLEU and ROUGE-L metrics. We have discussed the possible reasons for this drop in performance by CGenEx-con in section 5.5.1.

(4) Both CGenEx-Con and CGenEx-Fuse outperform the mBART-MT baseline by a margin of 1.65% and 0.80% in terms of accuracy and F1 score for TI task, respectively.

(5) When we compare CGenEx-Fuse and CGenEx-Fuse models, we observe CGenEx-Fuse model always outperforms the CGenEx-Fuse for both tasks in any settings. Like, mBART:CGenEx-fuse outperforms mBART:CGenEx-con with an improvement of 2.45% and 1.38% in ROUGE-L and BERTScore metrics for the SBG task, respectively. This observation establishes the efficacy of adding a commonsense aware encoder module in our proposed model.

(6) From Table 3, we can conclude that multitasking always performs better than single-task settings in all the variants of our proposed model and standard generative baselines. This observation establishes the benefit of multi-task learning, where two or more related tasks are solved simultaneously and help each other to improve individual performance. Table 4 shows classwise precision, recall, and F1 scores of the Target identification task generated by single-task and multi-task variants of our proposed model (mBART:CGenEx-fuse). From this table, we can observe that, except "Culture" target class, the multi-task model performs better for other classes than the single task model. Confusion matrices of single-task and multi-task variants of the mBART-CGenEx-fuse model for target identification task are shown in Figure 3.

We performed a statistical t-test on values of 5 runs of the proposed models and baseline models and obtained a p-value of = 0.005, which is less than 0.05 showing that the results
are statistically significant. We employ scipy library functions `stats.ttest_ind` for the t-test. We have highlighted (gray color) the results in Table 3 which are statistically significant.

Table 4: Classwise Precision, Recall, and F1 score of the Target identification task generated by single task and multitask variants of our proposed model (CGenEx-fuse)

<table>
<thead>
<tr>
<th>Class</th>
<th>Single Task</th>
<th>Multi-task</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1-score</td>
</tr>
<tr>
<td>Disabled</td>
<td>0.92</td>
<td>0.76</td>
<td>0.83</td>
</tr>
<tr>
<td>Race</td>
<td>0.91</td>
<td>0.84</td>
<td>0.88</td>
</tr>
<tr>
<td>Body</td>
<td>0.70</td>
<td>0.30</td>
<td>0.42</td>
</tr>
<tr>
<td>Victim</td>
<td>0.82</td>
<td>0.32</td>
<td>0.46</td>
</tr>
<tr>
<td>Gender</td>
<td>0.68</td>
<td>0.97</td>
<td>0.80</td>
</tr>
<tr>
<td>Culture</td>
<td>0.81</td>
<td>0.87</td>
<td>0.84</td>
</tr>
<tr>
<td>Social</td>
<td>0.88</td>
<td>0.29</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Figure 3: Confusion matrices: Single task Vs Multi-task variants of mBART-CGenEx-fuse model for target identification task.

5.3.1 Ablation Study
We performed an ablation study of our proposed models (CGenEX-con and CGenEx-fuse) to show the effect of reinforcement learning training (Table 5). It can be observed that removing the RL training from both variations of models results in a drop in performance in both tasks, Target classification, and Stereotype generation. Removing RL training from CGenEx-con results in a drop in the performance of 3.25% in the accuracy of the target classification task and 1.94% in BERTScore of the stereotype generation task. Similarly, removing RL training from CGenEx-fuse results in a drop in the performance of

https://docs.scipy.org/doc/scipy-1.6.3/reference/generated/scipy.stats.ttest_ind.html
Table 5. Ablation Study to show the effect of reinforcement learning based training

<table>
<thead>
<tr>
<th>Model</th>
<th>Target Acc</th>
<th>Target F1</th>
<th>Stereotype Bleu</th>
<th>Stereotype Rouge-L</th>
<th>Stereotype BERTScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>mBART:CGenEx-con</td>
<td>84.36</td>
<td>72.98</td>
<td>42.32</td>
<td>46.38</td>
<td>83.89</td>
</tr>
<tr>
<td>-RL</td>
<td>81.11</td>
<td>69.06</td>
<td>39.43</td>
<td>45.44</td>
<td>81.95</td>
</tr>
<tr>
<td>mBART:CGenEx-fuse</td>
<td>84.77</td>
<td>73.24</td>
<td>44.23</td>
<td>48.83</td>
<td>85.27</td>
</tr>
<tr>
<td>-RL</td>
<td>82.82</td>
<td>70.91</td>
<td>39.35</td>
<td>45.27</td>
<td>82.14</td>
</tr>
</tbody>
</table>

1.95% in the accuracy of the target classification task and 3.13% in the BERTScore of the stereotype generation task. This shows that RL training plays a vital role in improving the performance for both tasks as the BLEU-based reward function (Equation 10) encourages the model to generate an output sequence close to the golden sequence.

5.4 Performance on English SBIC Dataset

We have evaluated the English SBIC dataset to assess the effectiveness of our CGenEx model on English language. Table 6 shows that our proposed models, mBART:CGenEx-con and mBART:CGenEx-fuse, outperform the baseline models (GPT-1 and GPT-2) significantly in both single-task and multitask settings. Interestingly, the multitask variants of our models consistently outperform the single-task settings. This finding suggests that addressing both stereotype generation and targeted group identification together leads to improved performance in each individual task. It indicates a strong correlation between these tasks, where the performance on one task positively influences the performance on the other.


<table>
<thead>
<tr>
<th>Model</th>
<th>Implied statement (Stereotypes)</th>
<th>Target group category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>Rouge-L</td>
</tr>
<tr>
<td>SBF-GPT1</td>
<td>49.9</td>
<td>40.2</td>
</tr>
<tr>
<td>SBF-GPT2</td>
<td>49.6</td>
<td>43.5</td>
</tr>
<tr>
<td>ST-mBART:CGenEx-con</td>
<td>51.43</td>
<td>55.12</td>
</tr>
<tr>
<td>ST-mBART:CGenEx-fuse</td>
<td>53.67</td>
<td>56.04</td>
</tr>
<tr>
<td>MT-mBART:CGenEx-con</td>
<td>53.28</td>
<td>57.06</td>
</tr>
<tr>
<td>MT-mBART:CGenEx-fuse</td>
<td>55.16</td>
<td>58.74</td>
</tr>
</tbody>
</table>
5.5 Error Analysis

A detailed analysis of the results produced by the best-performing models on stereotype generation in both single-task and multitask settings identified several instances where the model can falter, some of which are discussed in Table 7.

1. **Irrelevant stereotype generation:** It can be seen from Table 7 that in a Single task setting, CGenEx-con models generate a stereotype for the post, but it doesn’t discuss the underlying implicit hate speech or sarcasm of the post.

2. **Wrong target group prediction:** It is also evident from Table 7 that the CGenEx-con model predicts the wrong target group for the post in both multi-task and single-task settings. The true target group was culture, but the model predicts it as race showing that models fail to distinguish between such closely related target groups.

3. **Wrong keyword generated:** CGenEx-fuse model is able to generate a very similar stereotype as to the true stereotype in the single-task setting. However, it replaces the keyword holocaust with mass destruction, which changes the context completely.

4. **Similar meaning but different tokens:** In multitask setting, we can see that CGenEx-con generates a stereotype that has a semantic overlap with the true stereotype illustrating why BERTScore has a consistently high value as compared to BLEU or ROUGE score as BERTScore measures the semantic overlap between two text embeddings.

5. **Multi-task outperforms single-task model:** Both CGenEx-fuse and CGenEx-con in the multi-task setting are able to generate the correct stereotype for the input post as compared to single task setting, showing that adding an additional task of target classification is helping the model to understand the underlying stereotype and bias better.

<table>
<thead>
<tr>
<th>Post</th>
<th>True stereotype</th>
<th>True target</th>
<th>Model</th>
<th>Predicted Stereotype</th>
<th>Predicted Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why do Jews stink? Because they are not falling in the bath again.</td>
<td>यहूदी बदबू क्यों करते हैं? क्योंकि वे बाढ़ से गांव में नहीं चलते हैं।</td>
<td>संस्कृति (Culture)</td>
<td>SBF-GPT [Sap et al. (2019)]</td>
<td>यहूदी लोग जल गये थे (संस्कृति)</td>
<td>संस्कृति (Culture)</td>
</tr>
<tr>
<td>Trans: Jews were burned</td>
<td>संस्कृति (Culture)</td>
<td>ST: CGenEx-con</td>
<td>Trans: Jews are gross (आतंक)</td>
<td>संस्कृति (Culture)</td>
<td></td>
</tr>
<tr>
<td>Trans: Sheds light on the genocide</td>
<td>ST: CGenEx-fuse</td>
<td>प्रलय के बारे में प्रकाश डालता है (संस्कृति)</td>
<td>Trans: sheds light on mass destruction (आतंक)</td>
<td>संस्कृति (Culture)</td>
<td></td>
</tr>
<tr>
<td>Trans: Discusses about genocide</td>
<td>MT: CGenEx-fuse</td>
<td>नरसंहार की बात करता है (आतंक)</td>
<td>Trans: sheds light on the genocide (संस्कृति)</td>
<td>संस्कृति (Culture)</td>
<td></td>
</tr>
</tbody>
</table>

5.5.1 Failure of CGenEx-con model in the Hindi Language

It can be observed that incorporating commonsense reasoning through Conceptnet with simple concatenation to input posts doesn’t improve the model’s performance. The reason for this can be attributed to the fact there is not any multilingual commonsense database available. To leverage ConceptNet for our dataset, we first translated the input post into English language and then applied the commonsense extraction module, and then again translated the generated commonsense reasoning to the Hindi language. As translation happens twice, there is a high chance of semantic loss during these two steps, which leads to ineffectual commonsense reasoning. To further bolster our argument, we conducted an
Table 8: Translation of Hindi posts to English and commonsense inference generated by Conceptnet from English to Hindi.

<table>
<thead>
<tr>
<th>#</th>
<th>Hindi Post</th>
<th>Translated English Post</th>
<th>English Commonsense</th>
<th>Translated Hindi commonsense</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>जब बीए कुछ ऐसा कहती है जो आपको परेशान करता है लेकिन आप कोशिश कर रहे हैं कि अब आप पागल न हों</td>
<td>When bae says somethin that pisses you off but you're trying to not to be crazy anymore</td>
<td>crazy is similar in meaning to brainsick</td>
<td>पागल दिमागी बीमारी के अरू में समान है</td>
</tr>
<tr>
<td>2</td>
<td>लोग आजकल अमेरिका में काले होने के लिए कितना मुश्किल है के बारे में बात करते हैं... लेकिन सफेद होने के बारे में क्या है? जैसे गन्दा, हम वास्तव में लगभग एक समान खेल के मैदान पर हैं, हमारे दादा-बाप के ऊपर में यह आसान था.</td>
<td>People talk about how hard it is to be black in America nowadays... but what about being white? Messy as it is, we're really on an almost equal playing field. As our grandparents it was easy.</td>
<td>black is related to minority</td>
<td>काला अल्पसंख्यक से समान है</td>
</tr>
<tr>
<td>3</td>
<td>बाइक और काले पुरुषों में क्या समानता है? புத்தகம</td>
<td>What do bikes and black men have in common?</td>
<td>black is distinct from white</td>
<td>काला सफेद से अलग है</td>
</tr>
<tr>
<td>4</td>
<td>holocaust के दौरान यहूदियों की एक गुप्त भाषा थी जो वे बोलते थे... यह मुझे इस तरह था, &quot;ऐ मेरे भगवान! मैं जीवन बर्बाद हूँ&quot;</td>
<td>During the holocaust the Jews had a secret language that they spoke... It was something like, &quot;Oh my God! I'm burning alive&quot;</td>
<td>jew is an ethnic group</td>
<td>यहूदी एक पंथ है</td>
</tr>
</tbody>
</table>

Error analysis to analyze and study the effect of these translations to better understand the semantic loss occurring while translating, which is shown in Table 8. It can be seen from the table that both translations happen correctly in the first two examples. However, in the third example, the first translation fails to translate the input Hindi post correctly as it misses the corresponding English word for Hindi word शृंखला, which completely changes the context of the input sentence. In the fourth example, the first translation happens correctly. However, the second translation (English commonsense to Hindi commonsense) fails as it mistranslated the word ethnic, which can misguide the model rather than help the model.

5.6 Limitations
In this work, we primarily focused on detecting and analyzing explicit hate speech in social media posts. Detecting sarcasm accurately in text is a complex task, as it often relies on contextual cues, tone, and understanding of cultural references. It goes beyond the scope of our current study, which primarily focuses on explicit and overt forms of hate speech. However, we acknowledge the significance of sarcasm as a potential element in hate speech and its impact on targeted groups. It is an important aspect to consider in future research and system development.

6. CONCLUSION AND FUTURE WORKS
As explainable AI systems help improve trustworthiness and confidence while deployed in real-time, now there is a need to explain why a post is predicted as hate by any model. To encourage more research on explainable hate speech detection in Hindi (the fourth-most-spoken language in the world), we introduced a Hindi hate speech dataset (HHES) that contains the stereotypical bias and target group category of a toxic post. In this
work, a unified generative framework (CGenEx) based on commonsense knowledge and
reinforcement learning has been proposed to simultaneously solve two tasks (stereotypical
bias generation and target group category identification). We showed how a multitasking
problem can be formulated as a text-to-text generation task to leverage the knowledge of
large pre-trained sequences to sequence models in low-resource language settings. Our
proposed model (CGenEx-fuse) outperforms the best baseline with an improved f1 score
of 0.80% and ROUGE-L of 2.09% for the target group identification and bias generation
tasks, respectively. We have also examined that the simple concatenation-based (CGenEx-
con) model is not performing as expected due to the semantic loss during English to Hindi
commonsense knowledge translation.

In our future work, we plan to investigate potential modifications to the CGenEx-con
model to address the challenges associated with English to Hindi commonsense knowledge
translation. These modifications may involve integrating language-specific semantic and
syntactic rules, utilizing bilingual resources and pre-trained models, or exploring transfer
learning techniques to enhance the quality of translation. Future attempts will be made to
extend explainable hate speech detection in a multimodal setting considering image and
text modality.

References


Badjatiya, P., Gupta, S., Gupta, M., and Varma, V. 2017. Deep learning for hate speech detection in


Cuddy, A. J., Fiske, S. T., Kwan, V. S., Glick, P., Demoulin, S., Leyens, J.-P., Bond, M. H.,
Croizet, J.-C., Ellemers, N., Sleebos, E., and others 2009. Stereotype content model across cultures:

Dadvar, M., Trieschnigg, D., and Jong, F. d. 2014. Experts and machines against bullies: A hybrid


Davidson, T., Warmsley, D., Macy, M., and Weber, I. 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, pp. 512–515.


Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., and Yang, G.-Z. 2019. Xai—explainable

Kamble, S. and Joshi, A. 2018. Hate speech detection from code-mixed hindi-english tweets using deep

Karim, M. R., Dey, S. K., Islam, T., Sarker, S., Menon, M. H., Hossain, K., Hossain, M. A.,
and Decker, S. 2021. DeepHateExplain: Explainable hate speech detection in under-resourced bengali
language. In 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA),
pp. 1–10. IEEE.

Kumar, R., Reganti, A. N., Bhatia, A., and Maheshwari, T. 2018. Aggression-annotated corpus of


