Natural Language Processing (2024), 1–20
doi:10.1017/nlp.2024.29

ARTICLE

StereoHate: Toward identifying stereotypical bias and target group in hate speech detection

Krishanu Maity[1], Nilabja Ghosh[2], Raghav Jain[1], Sriparna Saha[1] and Pushpak Bhattacharyya[3]

[1]Department of CSE, Indian Institute of Technology Patna, Patna, India, [2]Department of Computer Science, Ramakrishna Mission Vivekananda Educational and Research Institute, Howrah, India, and [3]Department of CSE, Indian Institute of Technology Bombay, Mumbai, India, India
Corresponding author: Krishanu Maity; Email: krishanumaity@gmail.com

(Received 25 December 2022; revised 3 June 2023; accepted 24 July 2023)

Special Issue on '**Natural Language Processing Applications for Low-Resource Languages**'

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 hate speech explanation (*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, *CGenEx,* 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 (*CGenEx-fuse*) on various evaluation metrics over other baselines when applied to the Hindi *HHES* dataset.

Disclaimer

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

Keywords: hate speech; stereotypical bias; explainability; multitask

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, Lee, and Hoang 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

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

natural language processing (NLP), numerous studies have suggested methods to detect hate speech automatically using traditional machine learning (ML) (Dinakar, Reichart, and Lieberman 2011; Reynolds, Kontostathis, and Edwards 2011; Dadvar, Trieschnigg, and Jong 2014) and deep learning approaches (Waseem and Hovy 2016; Badjatiya *et al.* 2017; Agrawal and Awekar 2018). 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 AI (Gunning *et al.* 2019) has necessitated the provision of explanations and interpretations for decisions made by ML 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 (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 (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 toward 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, Bhattacharya, and Weber 2019).

In the past decade, extensive research has been conducted to develop datasets and models for the automatic detection of online hate speech in the English language (Waseem and Hovy, 2016; Badjatiya et al., 2017; Agrawal and Awekar, 2018). 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 AI has now necessitated the provision of explanations and interpretations for decisions made by ML 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 the 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 (HHES) dataset that contains the stereotypical bias and target group category of a toxic post. To create the HHES dataset, we manually translate the existing English Social Bias Inference Corpus (SBIC) (Sap *et al.* 2020a) dataset. Now, we have to develop an efficient multitask (MT) framework that can solve two different categories of tasks simultaneously, that is, (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 cyberbullying detection) (Maity and Saha 2021b), complaint identification (Singh *et al.* 2022), and tweet act classification (Saha *et al.* 2022). A typical MT 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 (Crawshaw 2020) (where multiple tasks instead of optimizing

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

the learning process start to hurt the training process), model capacity (Wu 2019) (if the size of the shared encoder becomes too large, then there will be no transfer of information across different tasks), or optimization scheme (Wu 2019) (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 MT 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" and (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, that is, stereotypical bias/explanation (generation task) and identifying target group (classification task), a commonsense-aware unified generative framework (*CGenEx*) with reinforcement learning-based training has been proposed.[a]

The organization of this article is as follows. A survey of all the previous works in this domain is explained in Section 2. Section 3 describes the process of dataset creation in detail. Section 4 explains the proposed methodology, and Section 5 describes the experimental settings and results. This part also contains a detailed error analysis of our results.

2. 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.

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 percent in the F1 score compared to previous work that used statistical classifiers. The authors emphasized the ability of their models to capture the

[a] The code and dataset will be made publicly available in the camera-ready version.

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

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 and Saha (2021a) introduce 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 ML and deep neural network baselines. The model achieves an accuracy of 79.28 percent 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) address 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 term frequency-inverse document frequency (TF-IDF) methods by a significant margin of approximately 18 percent 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) present an approach for detecting hate speech on Twitter by leveraging patterns and unigrams collected from the training set as features for ML algorithms. The proposed method achieves high accuracies of 87.4 percent for binary classification (offensive vs. non-offensive) and 78.4 percent 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) address 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, Eisner, and Piatko (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 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 dataset collected from different social media like Twitter,

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

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 to identify important terms and generate human-interpretable explanations. Evaluations against ML 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 and 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 HHES (stereotypes) dataset. 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 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 make 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 1,010 million,[b] exhibits significant linguistic diversity. Among the numerous languages spoken in India,

[b] https://en.wikipedia.org/wiki/List_of_countries_by_number_of_Internet_users

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

Table 1. Train, validation, and test split distribution of target group category in HHES dataset

	Target group category							
Split	Race	Gender	Social	Body	Culture	Disabled	Victim	Total
Train	3462	3294	538	369	2134	713	1600	12110
Val	681	441	72	54	342	92	124	1806
Test	716	439	75	47	421	102	124	1924

Hindi holds a prominent position as one of the official languages,[c] with over 691 million speakers.[d] 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 HHES dataset. 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 one hundred 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 numbers of training, validation, and test samples in the HHES dataset are 12,110, 1,806, and 1,924, 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, Ekbal, and Bhattacharyya (2022). Fluency evaluates whether the translation is syntactically correct or not, whereas adequacy checks the semantic quality. Each annotator marked every translated sentence with an ordinal value from a scale of 1 to 5[e] for both F and A. We attain high average F and

[c] https://en.wikipedia.org/wiki/Hindi

[d] https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India

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

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

Table 2. Two examples from HHES dataset

Post	Target stereotype	Target group category
Hindi: मैं आज तुमसे लड़ने के लिए नहीं गया हूँ. मुझे वापस अपने निकगा हाथ दे English: What up fake ass hoe? I’m not gone fought you today. Gimme my nigga hand back”	निगा काल्पनिक शब्द काल्पनिक लोगों के लिए है। Nigga is a slang term for black people	Race
Hindi: और मैं एक कूड़ा-कूड़ा बदचलन के बारे में जानी नहीं जा सकती। English: And I can’t be known for fucking wit a trash bitch	औरतें बदचलन हैं Women are bitches	Gender

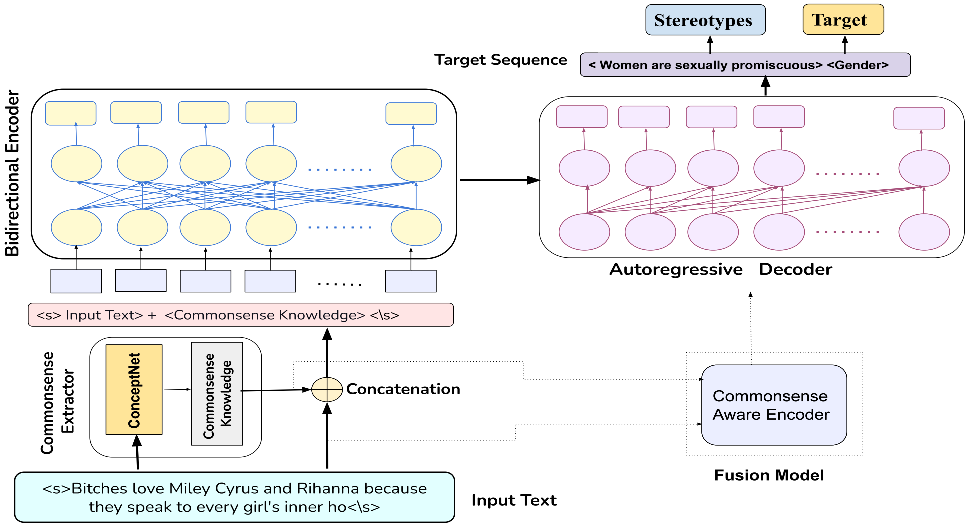

Stereotypes
Target
Target Sequence
< Women are sexually promiscuous> <Gender>
Bidirectional Encoder
Autoregressive Decoder
<s> Input Text> + <Commonsense Knowledge> <\s>
Commonsense Extractor
ConceptNet
Commonsense Knowledge
Concatenation
Commonsense Aware Encoder
Fusion Model
<s>Bitches love Miley Cyrus and Rihanna because they speak to every girl's inner ho<\s>
Input Text

Figure 1. A commonsense-aware unified generative framework *(CGenEx)* architecture.

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

$$Y_i = \{< St >< Tg >\} \tag{1}$$

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

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):

$$Y' = G(X) \tag{2}$$

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 pre-training 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 percent 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 dataset 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 the model can incorporate knowledge regarding social entities and events involved in the input text. We use ConceptNet (Speer, Chin, and Havasi 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 five 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 five 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 five triplets. To obtain the final commonsense reasoning CS for each input text, X_i, we concatenate these five commonsense reasonings together.

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

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 θ 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

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

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 the below equation:

$$H^t_{DEC} = G_{Decoder}(H_{EN}, Y_{<t}) \tag{4}$$

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^t_{DEC}, as follows:

$$P\left(Y'_t|X, Y_{<t}\right) = F_{softmax}\left(H^t_{DEC} W_{Gen}\right) \tag{5}$$

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 2), 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

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

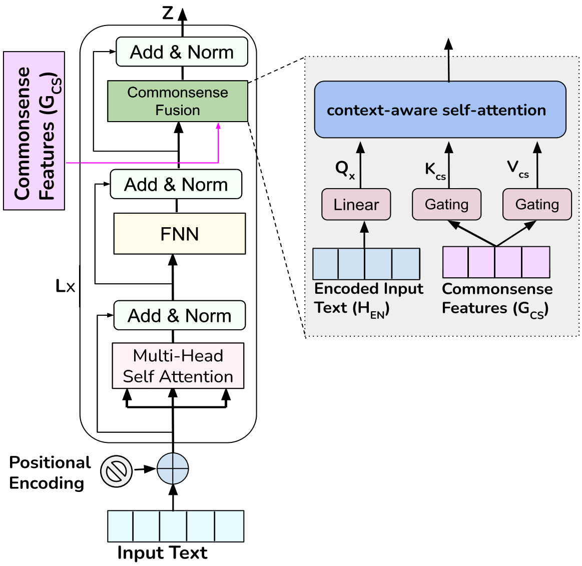

Z
Add & Norm
Commonsense Fusion
Commonsense Features (G_{CS})
Add & Norm
FNN
Lx
Add & Norm
Multi-Head Self Attention
Positional Encoding
Input Text
context-aware self-attention
Q_x
K_{cs}
V_{cs}
Linear
Gating
Gating
Encoded Input Text (H_{EN})
Commonsense Features (G_{CS})

Figure 2. Commonsense-aware encoder module internal architecture.

(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 λ_K and λ_V to create context-aware K_{cs} and V_{cs} (equation (6)):

$$\begin{bmatrix} K_{cs} \\ V_{cs} \end{bmatrix} = \left(1 - \begin{bmatrix} \lambda_K \\ \lambda_V \end{bmatrix}\right) \begin{bmatrix} K_x \\ V_x \end{bmatrix} + \begin{bmatrix} \lambda_K \\ \lambda_V \end{bmatrix} \left(G_{CS} \begin{bmatrix} U_K \\ U_V \end{bmatrix} \right) \tag{6}$$

where U_K and U_V are learnable parameters and matrices λ_K and λ_V are computed as follows:

$$\begin{bmatrix} \lambda_K \\ \lambda_V \end{bmatrix} = \sigma \left(\begin{bmatrix} K_x \\ V_x \end{bmatrix} \begin{bmatrix} W_K^X \\ W_V^X \end{bmatrix} + G_{CS} \begin{bmatrix} U_K \\ U_V \end{bmatrix} \begin{bmatrix} W_K^{CS} \\ W_V^{CS} \end{bmatrix} \right) \tag{7}$$

where W_K^X, W_V^X, W_K^{CS}, and W_V^{CS} all are learnable parameters and σ 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

$$Z = softmax\left(\frac{Q_x K_{cs}^T}{\sqrt{d_k}}\right) V_{cs} \tag{8}$$

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 θ 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, that is, the maximum likelihood estimation (MLE) objective function, which

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

works in a supervised manner to optimize the weights, θ, as defined in equation (9):

$$\max_{\theta} \prod_{t=0}^{T} P_{\theta}\left(Y_t' | X_i, Y_{<t}\right) \tag{9}$$

(2) On top of the 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.* 2002) based reward function. We define BLEU-based reward R_{BLEU} in equation (10):

$$R_{BLEU} = \left(BLEU\left(Y_i', Y_i\right) - BLEU\left(Y_i^g, Y_i\right)\right), \tag{10}$$

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).

$$\nabla_{\theta} J(\theta) = R_{BLEU} \cdot \nabla_{\theta} logP\left(Y_i' | X_i, CS; \theta\right) \tag{11}$$

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, that is, 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 groups 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 110,000 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[f] and PyTorch[g] as a backend. For the target category detection task, accuracy and macro-F1 metrics are used to evaluate predictive performance. For the stereotype generation task, we used BLEU (Papineni *et al.* 2002b), ROUGE-L (ROUGE, 2004), and BERTScore (Zhang *et al.* 2019).

(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 Lin and Och (2004).

[f] https://scikit-learn.org/stable/
[g] https://pytorch.org/

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

(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.

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 Mathew *et al.* (2020) for the target group identification task. BERT (Devlin *et al.* 2018) 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 128×768, is passed through 1D CNN layers. These layers consist of three kernel sizes (1, 2, 3) and one hundred 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 one hundred neurons, followed by an output softmax layer.
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.

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 MT 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 the 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.

Table 3. Results of different baselines and the two proposed frameworks, *CGenEx-con* and *CGenEx-fuse,* in a multitask 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

	Stereotype			Target		
Model	BLEU	ROUGE-L	BERTScore	Accuracy	F1 score	MCC
Standard baselines						
CNN-GRU	-	-	-	60.23	43.33	47.25
BiRNN	-	-	-	60.81	43.99	48.12
BiRNN+attention	-	-	-	62.33	44.31	50.37
BERT-finetune	-	-	-	65.41	47.37	52.69
mT5-ST	36.38	39.87	78.52	62.43	45.78	50.61
mT5-MT	37.58	41.14	79.72	64.12	46.88	52.19
mBART-ST	41.47	45.72	81.12	81.23	67.35	70.23
mBART-MT	42.25	46.33	82.28	83.12	72.44	71.84
Proposed model (CGenEx) single task						
mT5:CGenEx-con	37.14	40.89	79.12	65.53	47.84	55.31
mT5:CGenEx-fuse	38.62	42.37	80.46	66.92	48.07	55.93
mBART:CGenEx-con	41.87	45.93	82.77	83.54	72.66	74.52
mBART:CGenEx-fuse	42.95	46.74	83.09	83.83	72.78	74.72
Proposed model (CGenEx) multitask						
mT5:CGenEx-con	38.25	42.14	80.75	67.96	48.67	56.76
mT5:CGenEx-fuse	38.88	42.97	81.63	68.12	48.86	56.83
mBART:CGenEx-con	43.12	47.08	83.89	84.36	72.98	75.86
mBART:CGenEx-fuse	**44.23**	**48.83**	**85.27**	**84.77**	**73.24**	**76.26**

(2) When we compare the performance of generative baselines, mBART always performs better than mT5 in both single-task (ST) and MT settings. Like, mBART-ST outperforms the mT5-ST model by a margin of (1) 18.80 percent and 21.57 percent in accuracy and F1 score for TI task, respectively, and (2) 5.09 percent, 5.85 percent, and 2.60 percent in BLEU, ROUGE-L, and BERTScore metrics for the 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 percent and 2.09 percent in BLEU and ROUGE-L metrics for the SBG task, respectively, and (ii) 0.94 percent 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:

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)

	Single task			Multitask			
Class	Precision	Recall	F1 score	Precision	Recall	F1 score	Support
Disabled	0.92	0.76	0.83	0.99	0.73	**0.84**	102
Race	0.91	0.84	0.88	0.89	0.89	**0.89**	716
Body	0.70	0.30	0.42	0.77	0.43	**0.55**	47
Victim	0.82	0.32	0.46	0.41	0.66	**0.51**	124
Gender	0.68	0.97	0.80	0.85	0.82	**0.84**	439
Culture	0.81	0.87	**0.84**	0.80	0.84	0.82	421
Social	0.88	0.29	0.44	0.79	0.51	**0.62**	75

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 percent and 0.80 percent 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 percent and 1.38 percent 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 multitask 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 multitask variants of our proposed model (mBART:CGenEx-fuse). From this table, we can observe that except "culture" target class, the multitask model performs better for other classes than the single-task model. Confusion matrices of single-task and multitask 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*[h] for the t-test. We have highlighted (gray color) the results in Table 3 which are statistically significant.

5.3.1. Ablation study

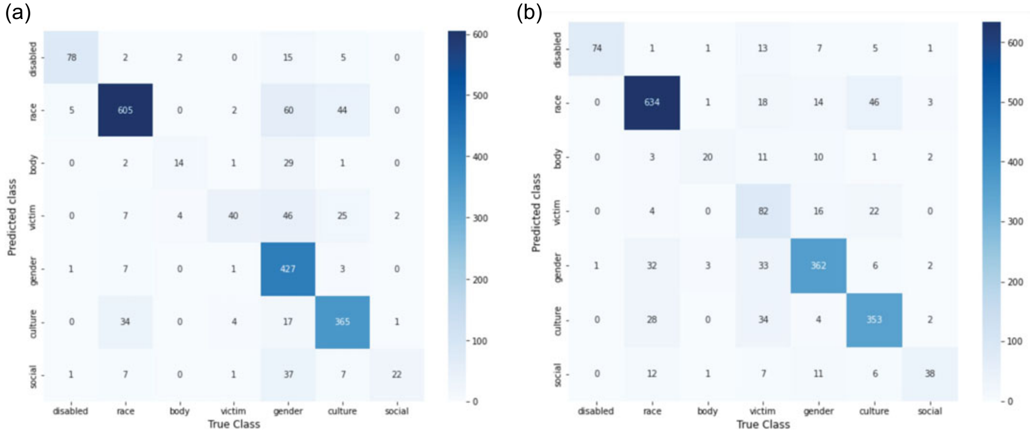
We performed an ablation study of our proposed models (CGenEX-con and CGenEx-fuse) to show the effect of reinforcement learning training (Table 5). It can be observed that removing the reinforcement learning (RL) training from both variations of models results in a drop in performance in both tasks,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

[h] https://docs.scipy.org/doc/scipy-1.6.3/reference/generated/scipy.stats.ttest_ind.html

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

Table 5. Ablation study to show the effect of reinforcement learning-based training

	Multitask setting				
	Target		Stereotype		
Model	Acc	F1	Bleu	Rouge-L	BERTScore
mBART:CGenEx-con	84.36	72.98	42.32	46.38	83.89
-RL	81.11	69.06	39.43	45.44	81.95
mBART:CGenEx-fuse	**84.77**	**73.24**	**44.23**	**48.83**	**85.27**
-RL	82.82	70.91	39.35	45.27	82.14



Single task

Mutitask

Figure 3. Confusion matrices: single-task vs. multitask variants of mBART-CGenEx-fuse model for target identification task.

removing RL training from CGenEx-fuse results in a drop in the performance of 1.95 percent in the accuracy of the target classification task and 3.13 percent 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 the 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 MT settings. Interestingly, the MT 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.

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

Table 6. Comparison of performance on English SBIC dataset: proposed models vs. baseline models in single-task and multitask settings

	Task			
	Implied statement (stereotypes)		Target group category	
Model	BLEU	Rouge-L	Acc	F1
SBF-GPT1 Sap *et al.* (2019b)	49.9	40.2	-	-
SBF-GPT2 Sap *et al.* (2019b)	49.6	43.5	-	-
ST-mBART:CGenEx-con	51.43	55.12	82.76	72.08
ST-mBART:CGenEx-fuse	53.67	56.04	86.23	75.14
MT-mBART:CGenEx-con	53.28	57.06	83.98	72.85
MT-mBART:CGenEx-fuse	55.16	58.74	87.74	76.39

Table 7. Comparative study of stereotype and target of a post by proposed models vs. actual annotations; ST: single task, MT:multitask; mBART embedded models have been selected for error analysis

Post	True stereotype	True target	Model	Predicted stereotype	Predicted target
यहूदी बदबू क्यों करते हैं? क्योंकि वे फिर से नहाने के लिए नहीं गिर रहे हैं। Translation: Why do Jews stink? Because they are not falling in the bath again.	नरसंहार के बारे में प्रकाश डालता है Trans: Sheds light on the genocide	संस्कृति (Culture)	SBF-GPT Sap *et al.* (2019b)	यहूदी लोग जल गये थे Trans: Jewish folks were burned	संस्कृति (Culture)
			ST:CGenEx-con	यहूदियों का गंध Trans: Jews are gross	जाति (Race)
			ST:CGenEx-fuse	प्रलय के बारे में प्रकाश डालता है Trans: sheds light on mass destruction	संस्कृति (Culture)
			MT:CGenEx-con	नरसंहार की बात करता है Trans: discusses about genocide	जाति (Race)
			MT:CGenEx-fuse	नरसंहार के बारे में प्रकाश डालता है Trans: sheds light on the genocide	संस्कृति (Culture)

5.5. *Error analysis*

A detailed analysis of the results produced by the best-performing models on stereotype generation in both single-task and MT 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 multitask 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.

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

Table 8. Translation of Hindi posts to English and commonsense inference generated by ConceptNet from English to Hindi

#	Hindi post	Translated English post	English commonsense	Translated Hindi commonsense
1	जब बीए कुछ ऐसा कहती है जो आपको परेशान करता है लेकिन आप कोशिश कर रहे हैं कि अब आप पागल न हों	When bae says somethin that pisses you off but you're trying to not to be crazy anymore	Crazy is similar in meaning to brainsick	पागल दिमागी बीमारी के अर्थ में समान है
2	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 रूप में यह आसान था.	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.	Black is related to minority	काला अल्पसंख्यक से संबंधित है
3	बाइक और काले पुरुषों में क्या समानता है? शृंखला	What do bikes and black men have in common?	Black is distinct from white	काला सफेद से अलग है
4	holocaust के दौरान यहूदियों की एक गुप्त भाषा थी जो वे बोलते थे... यह कुछ इस तरह था, "ऐ मेरे भगवान! मैं जीवित बर्निंग हूँ"	During the holocaust the Jews had a secret language that they spoke. . . It was something like, "Oh my God! I'm burning alive"	Jew is an ethnic group	यहूदी एक पंथ है

(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 the MT 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) Multitask outperforms single-task model: Both CGenEx-fuse and CGenEx-con in the multitask 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.

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 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.

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

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 HHES dataset 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 percent and ROUGE-L of 2.09 percent 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.

Acknowledgments. Dr Sriparna Saha gratefully acknowledges the Young Faculty Research Fellowship (YFRF) Award, supported by Visvesvaraya PhD Scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia) for carrying out this research.

References

Agrawal S. and Awekar A. (2018). Deep learning for detecting cyberbullying across multiple social media platforms. In *European conference on information retrieval*, Springer, pp. 141–153.

Badjatiya P., Gupta M. and Varma V. (2019). Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *The World Wide Web Conference*, pp. 49–59.

Badjatiya P., Gupta S., Gupta M. and Varma V. (2017). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 759–760.

Cao R., Lee R. K.-W. and Hoang T.-A. (2020). Deephate: Hate speech detection via multi-faceted text representations. In *12th ACM conference on web science*, pp. 11–20.

Caruana R. (1997). Multitask learning. In *Machine Learning*, **28**.

Crawshaw M. (2020). Multi-task learning with deep neural networks: A survey.

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., et al. (2009). Stereotype content model across cultures: Towards universal similarities and some differences. *British Journal of Social Psychology* **48**(1), 1–33.

Dadvar M., Trieschnigg D. and Jong F.d (2014). Experts and machines against bullies: A hybrid approach to detect cyberbullies. In *Canadian conference on artificial intelligence*, Springer, pp. 275–281.

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

Davidson T., Bhattacharya D. and Weber I. (2019). Racial bias in hate speech and abusive language detection datasets.arXiv preprint arXiv: 1905.

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*, vol. **11**, pp. 512–515.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv: 1810.04805.

Dinakar K., Reichart R. and Lieberman H. (2011). Modeling the detection of textual cyberbullying. In *Proceedings of the International Conference on Weblog and Social Media 2011*, Citeseer.

Ghosh S., Ekbal A. and Bhattacharyya P. (2022). Am i no good? *towards Detecting Perceived Burdensomeness and Thwarted Belongingness From Suicide Notes*, arXiv preprint arXiv: 2206.06141.

Gunning D., Stefik M., Choi J., Miller T., Stumpf S. and Yang G.-Z. (2019). Xai–explainable artificial intelligence. *Science Robotics* **4**(37), eaay7120.

Kamble S. and Joshi A. (2018). Hate speech detection from code-mixed Hindi-english Tweets using deep learning models, arXiv preprint arXiv: 1811.05145.

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

Kumar R., Reganti A. N., Bhatia A. and Maheshwari T. (2018). Aggression-annotated corpus of Hindi-english code-mixed data, arXiv preprint arXiv: 1803.09402.

Lewis M., Liu Y., Goyal N., Ghazvininejad M., Mohamed A., Levy O., Stoyanov V. and Zettlemoyer L. (2020). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, pp. 7871–7880.

Lin C.-Y. and Och F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 605–612.

Liu Y., Gu J., Goyal N., Li X., Edunov S., Ghazvininejad M., Lewis M. and Zettlemoyer L. (2020). Multilingual denoising pre-training for neural machine translation.

Ma X., Sap M., Rashkin H. and Choi Y. (2020). Powertransformer: Unsupervised controllable revision for biased language correctionarXiv preprint arXiv: 2010.13816.

Maity K. and Saha S. (2021a). Bert-capsule model for cyberbullying detection in code-mixed indian languages. In *International Conference on Applications of Natural Language to Information Systems*, Springer, pp. 147–155.

Maity K. and Saha S. (2021b). A multi-task model for sentiment aided cyberbullying detection in code-mixed Indian languages. In *International Conference on Neural Information Processing*, Springer, pp. 440–451.

Mathew B., Saha P., Yimam S. M., Biemann C., Goyal P. and Mukherjee A. (2020). Hatexplain: A benchmark dataset for explainable hate speech detection, arXiv preprint arXiv: 2012.10289.

Nockleby J. T. (1994). Hate speech in context: The case of verbal threats. *Buffalo Law Review* **42**, 653.

Papineni K., Roukos S., Ward T. and Zhu W.-J. (2002a). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–318.

Paul S. and Saha S. (2020). Cyberbert: Bert for cyberbullying identification. In*Multimedia Systems*, pp. 1–8.

Radford A., Wu J., Child R., Luan D., Amodei D. and Sutskever I. (2019). Language models are unsupervised multitask learners.

Raffel C., Shazeer N., Roberts A., Lee K., Narang S., Matena M., Zhou Y., Li W. and Liu P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* **21**(140), 1–67.

Regulation P. (2016). Regulation (EU) 2016/679 of the European parliament and of the council. *Regulation (EU)* **679**, 2016.

Reynolds K., Kontostathis A. and Edwards L. (2011). Using machine learning to detect cyberbullying. In *2011 10th International Conference on Machine Learning and Applications and Workshops*, IEEE, vol **2**, pp. 241–244.

Rosenfeld A. and Richardson A. (2019). Explainability in human–agent systems. *Autonomous Agents and Multi-Agent Systems* **33**(6), 673–705.

ROUGE, L. C. (2004). A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL*, Spain.

Saha T., Upadhyaya A., Saha S. and Bhattacharyya P. (2022). A multitask multimodal ensemble model for sentiment-and emotion-aided tweet act classification. *IEEE Transactions on Computational Social Systems* **9**(2), 508–517.

Sancheti A., Krishna K., Srinivasan B. and Natarajan A. (2020). Reinforced rewards framework for text style transfer.

Sap M., Card D., Gabriel S., Choi Y. and Smith N. A. (2019a). The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1668–1678.

Sap M., Gabriel S., Qin L., Jurafsky D., Smith N. A. and Choi Y. (2019b). Social bias frames: Reasoning about social and power implications of language, arXiv preprint arXiv: 1911.

https://doi.org/10.1017/nlp.2024.29 Published online by Cambridge University Press

Sap M., **Gabriel S.**, **Qin L.**, **Jurafsky D.**, **Smith N. A. and Choi Y.** (2020a). Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics, pp. 5477–5490.

Sap M., **Shwartz V.**, **Bosselut A.**, **Choi Y. and Roth D.** (2020b). Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, Online. Association for Computational Linguistics, pp. 27–33.

Singh A., **Saha S.**, **Hasanuzzaman M. and Dey K.** (2022). Multitask learning for complaint identification and sentiment analysis. *Cognitive Computation* **14**, 212–227.

Speer R., **Chin J. and Havasi C.** (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence.*

Sridhar R. and Yang D. (2022). Explaining toxic text via knowledge enhanced text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 811–826.

Vaswani A., **Shazeer N.**, **Parmar N.**, **Uszkoreit J.**, **Jones L.**, **Gomez A. N.**, **Kaiser L.u and Polosukhin I.** (2017). Attention is all you need. In **Guyon I.**, **Luxburg U. V.**, **Bengio S.**, **Wallach H.**, **Fergus R.**, **Vishwanathan S. and Garnett R.**, (eds), *Advances in Neural Information Processing Systems*, **30**, Curran Associates, Inc.

Wang L., **Li R.**, **Yan Y.**, **Yan Y.**, **Wang S.**, **Wu W. and Xu W.** (2022). Instructionner: A multi-task instruction-based generative framework for few-shot ner, arXiv preprint arXiv: 2203.03903.

Waseem Z. and Hovy D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pp. 88–93.

Watanabe H., **Bouazizi M. and Ohtsuki T.** (2018). Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE Access* **6**, 13825–13835.

Wu S. (2019). Emmental: A framework for building multimodal multi-task learning systems.

Yan H., **Dai J.**, **Ji T.**, **Qiu X. and Zhang Z.** (2021). A unified generative framework for aspect-based sentiment analysis. *CoRR*, abs/2106.04300.

Yang B., **Li J.**, **Wong D.**, **Chao L.**, **Wang X. and Tu Z.** (2019). Context-aware self-attention networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. **33**, pp. 387–394.

Zaidan O., **Eisner J. and Piatko C.** (2007). Using "annotator rationales" to improve machine learning for text categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 260–267.

Zhang T., **Kishore V.**, **Wu F.**, **Weinberger K. Q. and Artzi Y.** (2019). Bertscore: Evaluating text generation with bert, arXiv preprint arXiv: 1904.

Cite this article: Maity K, Ghosh N, Jain R, Saha S and Bhattacharyya P. StereoHate: Toward identifying stereotypical bias and target group in hate speech detection. *Natural Language Processing* https://doi.org/10.1017/nlp.2024.29