Proactively Reducing the Hate Intensity of Online Posts via Hate Speech Normalization

Sarah Masud*, Manjot Bedi‡, Mohammad Aflah Khan*, Md Shad Akhtar*, Tanmoy Chakraborty*

*Dept. of CSE, IIIT-Delhi, India; ‡Northeastern University, USA
{sarahm,aflah20082,shad.akhtar,tanmoy}@iiitd.ac.in,bedi.m@northeastern.edu

ABSTRACT
Curbing online hate speech has become the need of the hour; however, a blanket ban on such activities is infeasible due to several geopolitical and cultural reasons. To reduce the severity of the problem, in this paper, we introduce a novel task, hate speech normalization, which aims to weaken the intensity of hatred exhibited by an online post. The intention of hate speech normalization is not to support hate but instead to provide the users with a stepping stone towards non-hate while giving online platforms more time to monitor any improvement in the user’s behavior.

To this end, we manually curated a parallel corpus – hate texts and their normalized counterparts (a normalized text is less hateful and more benign). We introduce NACL, a simple yet efficient hate speech normalization model that operates in three stages – first, it measures the hate intensity of the original sample; second, it identifies the hate span(s) within it; and finally, it reduces hate intensity by paraphrasing the hate spans.1 We perform extensive experiments to measure the efficacy of NACL via three-way evaluation (intrinsic, extrinsic, and human-study). We observe that NACL outperforms six baselines – NACL yields a score of 0.1365 RMSE for the intensity prediction, 0.622 F1-score in the span identification, and 82.27 BLEU and 80.05 perplexity for the normalized text generation. We further show the generalizability of NACL across other platforms (Reddit, Facebook, Gab). An interactive prototype of NACL was put together for the user study. Further, the tool is being deployed in a real-world setting at Wipro AI as a part of their mission to tackle harmful content on online platforms.

CCS CONCEPTS
• Computing methodologies → Natural language generation; Neural networks; • Information systems → Social networks.

KEYWORDS
Online Hate Speech, Hate Intensity Reduction, Span Detection, Hate Normalization, Proactive Strategy, Cross-Platform Tool.

1 Disclaimer: This paper includes examples of hate speech which contain some profane words. These examples are only included for contextual understanding. We tried our best to censor vulgar, offensive, or hateful words. We assert that we do not support these views in any way.

1 INTRODUCTION

Today, anyone with access to the Internet can learn, share, and participate in matters that concern them. However, anti-social behavior such as spreading fake news, cyberbullying, polarization, and posting hateful messages has increased with Internet penetration. While what constitutes hate speech is subjective, there is a common understanding of what underpins hate [16]. Hate speech can be understood as any communication in speech, writing, or behavior that attacks/discriminates or uses pejorative language against an individual or a group based on who they are. The attack is often targeted at religion, ethnicity, nationality, race, color, descent, gender, or other identity factors [47]. Our experiment aims to capture varying degrees and designs of hatefulness; we use hate as an umbrella term encompassing overlapping definitions of hate, abuse, and offense. Hate speech on social media usually consists of content that is often in violation of the terms of usage of the platform and leads to either flagging of content or banning of users or both. This has led to an influx of banned users turning over to less-restrictive platforms [15]. A cross-platform study of hateful users revealed how the banning of users could backfire for the platform and the Internet community at large. [27].

Motivation and Evidence. An alternative solution is to proactively counter [7] hate speech before it goes public. Systematic and proactive sensitization of online users can help them voice their opinions without directly propagating harm. In this case, we believe that promoting users with alternative scenarios nudges a change in attitude [5, 29]. Twitter and Instagram, among other platforms, have already rolled out the experimental prompts for offensive posts with an option to revise before going public’ [23, 24]. These tools state that the material is objectionable and leave it to the users to build a more appropriate form. Based on experiments with such prompts, researchers at Twitter published findings [28] that prompted participants to eventually post fewer offensive tweets than non-prompted users in the control group. Such nudging should be more welcoming to the users than them getting banned from the platform without having an opportunity to improve. 

Contribution I: Novel problem statement. Prompting users to change their texts to be completely non-hateful can be a significant behavioral shift and may backfire. Therefore, as an experiment, we take the first step of making the content less hateful and hope to build from there. Extending upon the idea of proactive prompts, we propose the novel task of hate speech normalization, which suggests a normalized counterpart of hateful content by weakening the overall hatred exhibited by the original content while retaining the underlying semantics. Additionally, in line with previous studies [28], we observe a significant reduction in user engagement (virality) for normalized sentences compared to the original one. Our hypothesis testing is detailed in Section 2.
We perform an extrinsic evaluation on three standard state-of-the-reduced hate sample compared to the original one. We further compare the models in extrinsic evaluation, and coherent and better results in intrinsic evaluation, higher reduction in confidence outputs of hate detection models, and text normalization, respectively. We hypothesize that these models in classifying the normalized samples as 'hateful' against the original unnormalized samples. It results with various baselines. We develop an end-to-end system to normalize the identified hate spans. In the last stage, we incorporate and fine-tune a BART-based sequence-to-sequence model (seq2seq) to normalize the identified hate spans. Following this, a BiLSTM-CRF module operates on the normalized sample by employing an attention-driven BiLSTM-based regressor, which operates in three stages. It first measures the intensity of hatred exhibited by an incoming post and then normalize them to reduce overall hatred. Table 1 lists examples of such posts and their normalized counterparts.

Contribution II: New dataset. Note that our aim is not to render a hateful message into a non-hateful one; instead, it is the reduction of hatred. To this end, we manually curate a parallel corpus of hateful samples and their normalized versions, the first dataset of its kind. Our preliminary analysis of hateful samples suggests that, in general, only a few phrases within a sentence convey major hatred. Therefore, we first identify the hateful spans for each hateful sample and then normalize them to reduce overall hatred. Table 1 shows examples of such posts and their normalized counterparts.

Contribution III: Novel method. We then propose NACL, a Neural hAtE specCh normaLizer, which operates in three stages. It first measures the intensity of hatred exhibited by an incoming sample by employing an attention-driven BiLSTM-based regression model. Following this, a BiLSTM-CRF module operates on the sample to extract the hateful spans from the source sample. In the last stage, we incorporate fine-tune a BART-based sequence-to-sequence model (seq2seq) to normalize the identified hate spans with feedback (reward/penalty) from the hate intensity predictor-driven discriminator module. The reward/penalty from the discriminator enforces the generator (BART model) to paraphrase the hate spans to reduce the hate intensity. For each stage, we compare our results with various baselines. We develop an end-to-end system that can be efficiently deployed in the real-world environment. It achieves consistent performance in offline (intrinsic/extrinsic evaluations) as well as online evaluations (as observed by annotators).

Contribution IV: Extensive evaluation and deployment. We perform an extrinsic evaluation on three standard state-of-the-art hate detection models where we measure the confidence of these models in classifying the normalized samples as 'hate' against the original unnormalized samples. We hypothesize that the models should have lower 'hate' confidence for the normalized (reduced hate) sample compared to the original one. We further compare NACL with six baselines and obtain a performance of 0.136 RMSE, 0.622 F1-score, and 82.27 BLEU in intensity prediction, span identification, and text normalization, respectively.

In partnership with Wipro AI, we develop an interactive web interface and conduct an anonymized human evaluation on the outputs of NACL and other baselines. We observe that NACL reports better results in intrinsic evaluation, higher reduction in confidence of hate detection models in extrinsic evaluation, and coherent and better outputs as evaluated by humans. We further show the generalizability of NACL across three social media platforms – Reddit, Facebook, and Gab.

Reproducibility: Being aware of the ethical considerations (Appendix A) of dataset, we are restricting its release. Meanwhile, the models, source codes, annotation guidelines, the questionnaires used for human evaluation (c.f. Section 6.5), and web-tool (c.f. Section 6.7) is available at https://github.com/LCS2-IIITD/Hate_Norm.

2 APPLICATION AND TARGET AUDIENCE

Target Audience. People initiate and engage in hateful conversation for a variety of reasons. There will always be users who intentionally and constantly spread hateful content. The majority of such users eventually get flagged and reported by the community for their anti-social behavior. However, despite intentionally engaging in hateful content, a section of online users are adaptive and can be nudged to change their opinions via empathy. Our tool is aimed at the latter set of user groups whose social engagements can be eventually guided to become non-hateful. Meanwhile, prompting tools and the adoption of suggestive texts can help the content moderators understand the linguistic behavior of offensive users and track any changes in user behavior against getting flagged/banned.

Evidence. A significant criticism for prompting and suggestive rectification is the debate of reduced expressive powers of the users. However, a recent study by Twitter [28] also revealed that encouraging users to reduce the publishing of offensive text had no significant impact on their ability to participate in the non-offensive conversation. This result is a big motivation for researchers exploring proactive methods of countering hateful/hurtful content, as we do in this paper. Though one can argue that normalized text is still hateful and prone to spreading harm, we hypothesize that its normalized form with a reduction in intensity, the content should see a decline in the user engagement it receives (i.e., reduces its virality). We take inspiration from the virality prediction models on social media to test this hypothesis. In our setting, the virality of a post is expressed in terms of the total comments it receives. Since ours is a text-only dataset curated from different sources, we lack the availability of any network, temporal and other signals. Engineering various textual features (listed in Appendix B) inspired...
To achieve this, we divide the overall task into three stages.

• **Hate Intensity Prediction (HIP).** The task of hate intensity prediction is to determine the extent (degree or intensity) of hatred in a message (inspired by the idea of measuring online toxicity\(^\text{3}\)). Given a sample \(t\), it measures the intensity of hatred on a scale of \([1, 10]\), with 10 being the highest, i.e., \(\phi_t = HIP(t)\).

• **Hateful Span Identification (HSI).** Hateful spans in a sentence are the portions of a sentence responsible for conveying hate [38]. The intuition behind this task is that we can achieve sentence-level normalization if we attempt to normalize these spans. A sample can have multiple, non-overlapping hate spans, and \(HSI(t)\) aims to find all such spans.

• **Hate Intensity Reduction (HIR).** The objective of hate intensity reduction is to generate a new sample that preserves the original semantic but with weaker hate intensity. Intuitively, we first need to identify the threshold for strong vs weak hate. For our experiments, we set a threshold \(\tau = 5\). For each strong hate sample with \(\phi_t > \tau\), we aim to generate semantically similar sample \(t' \approx t\), with the constraint \(\phi_t' \leq \tau\).

![Figure 1: Difference in predicted number of comments per set per iteration. During each iteration, we obtain the median difference (\(\Delta\)) in the predicted number of comments for hateful and corresponding normalized sampled pairs.](https://www.perspectiveapi.com)

![Figure 2: A schematic view of NACL. The first stage of the model (i.e., HIP) validates the eligibility of a sample for normalization. Subsequently, HSI identifies the start and the end indices to mark hate spans in eligible samples. Finally, for each identified hate span, a normalized paraphrase is generated by HIR and validated by the discriminator to ensure the intensity reduction.](https://www.perspectiveapi.com)

### 3 OBJECTIVE & INDIVIDUAL STAGES

For a given hate sample \(t\), our objective is to obtain its normalized form \(t'\) such that the intensity of hatred \(\phi_t\) is reduced i.e., \(\phi_{t'} < \phi_t\). To achieve this, we divide the overall task into three stages.

- **HIP Module:** For hate intensity prediction, we employ a Bi-LSTM model. The intermediate representation is then passed through a self-attention layer connected to a fully-connected layer with linear activation to obtain the input sample’s continuous hate intensity prediction. Table 3 shows various models and pre-trained embeddings we experimented with.

- **HSI Module:** For hate intensity prediction and normalization, we employ a Bi-LSTM model. The intermediate representation is then passed through a self-attention layer connected to a fully-connected layer with linear activation to obtain the input sample’s continuous hate intensity prediction. Table 3 shows various models and pre-trained embeddings we experimented with.

- **NACL-HIP Module:** If a sample is hateful, we mimic it by only using hateful samples for training NACL (c.f. Section 5). Any future reference of the input sample would mean that it is already labeled as hate. In the first step, we compute the hate intensity (\(\phi_t\)) of an input sample \(t\) and forward it for normalization only if it satisfies the condition, \(\phi_t > \tau\) (i.e., only strongly hateful samples are subject to normalization). Next, the model predicts the hate spans within the incoming sample following the CONLL-2002 BIO notations [45]. Subsequently, NACL normalizes each identified hate span through a generator module. Finally, the outputs of each generator are interleaved in the original span sequence and fed to the discriminator to ensure the reduction of hate intensity (\(\phi \leq \tau\)). The pseudo-code is summarized in Appendix D.

3https://www.perspectiveapi.com
where \( \ell \) we experiment with two threshold values (\( s \) ranges across different sequences and \( x \) represents the POS tag). For our model, we use bidirectional LSTMs to capture the contextual representation of sequences. The hidden representations are then passed through a Time Distributed dense layer to flatten the embedding structure. We further use a CRF model layer to fit our representations and produce the required tags for our sequences. The CRF layer models the conditional probability of each sequence \( p(s_1, ..., s_m| x_1, ..., x_m) \) by defining a feature map \( \phi(x_1, ..., x_m, s_1, ..., s_m) \in R^d \) that maps an entire sequence to a d-dimensional feature vector. Then we can model the probability using the feature vector \( w \in R^d \) as follows:

\[
p(s|x; w) = \sum_s \exp(w, \phi(x, s)),
\]

where \( s \) ranges across different sequences and \( x \) represents the POS tag. After getting the feature vector \( w^* \), we can find the most likely tag for a sentence \( s^* \) by: \( s^* = \text{argmax} p(s|x; w^*) \)

\textbf{NACL-HIR Module}: The hate intensity reduction model is a Generative Adversarial Network (GAN) based architecture, which employs both NACL-HIP and NACL-HSI as its assistive modules. NACL-HIR accepts a hateful sample along with its span labels as identified by the HSI model. For NACL-HIR, we use a pre-trained BART [31] and fine-tune it for the hate normalization task. The BART-based seq-to-seq model generates the normalized spans based on the hate spans identified by the NACL-HSI module. The generated spans are amalgamated with the rest of the sample tokens and forwarded to the HIP-based discriminator model.

The discriminator computes the hate intensity score \( \phi_{\tau} \) of the normalized text \( t' \). Since our objective is to reduce the intensity of \( t' \), we assign a reward/penalty \( R \) to the generator based on the hate intensity score \( \phi_{\tau} \) of the normalized sentence and the accepted threshold value \( \tau \) as follows:

\[
R_{\tau} = \tau - \phi_{\tau}
\]

Note that the threshold value is a hyperparameter. In this paper, we experiment with two threshold values (\( \tau = \{3, 5\} \)) for the hate normalization. Table 5 shows that \( \tau = 5 \) is better suited for the NACL-HIR model. Therefore, for the rest of the experimentation, we continued with \( \tau = 5 \) as the threshold value for our model. If the computed hate intensity score is lesser than or equal to the threshold, the positive reward encourages the generator to continue predicting the similar normalized span; otherwise, the negative reward penalizes the generator to improve its prediction. The generator consumes the reward into its loss function as follows:

\[
L = \ell + (1 - R)
\]

where \( \ell \) is the generator loss. Backpropagation aims to minimize the consolidated loss \( L \) in order to generate semantically coherent normalized samples with hate intensity lesser than or equal to \( \tau \).

\textbf{Some Engineering Observations}. As stated at the beginning of this section, we aim to find a solution that not only performs well for the given problem statement but can also be deployed. Subsequently, we make a trade-off between efficiency and complexity at each modeling stage. For instance, adding additional BiLSTM layers for HIP did not significantly boost the performance; hence, we exclude them from the final modeling. Additionally, with HSI, simpler embedding solutions like Glove and ELMo performed comparably to BERT. We think the main reason for BERT’s low performance is that the BERT tokenizer splits the words into subtokens, which causes the spans to be distributed unequally across the subtokenized words. This skews the target values that belong to one of the tags (\( q_i \in (B, I, O) \)), resulting in the model overfitting. As we have limited data, the ELMo model performs better with this data and performs well for the problem. For both HSI and HIR, we make the obvious decision to use the optimized, distilled versions of the larger BERT and BART models (as supported by HuggingFace\(^3\)), further reducing the number of parameters to train.

\section{5 DATASET}

We sourced and compiled a list of hateful instances from [3, 9, 11, 12, 25, 35] by restricting our data creation to the hateful labels of the respective datasets. In total, we collected 4423 hateful samples and annotated them with hate intensity scores and hateful spans. At the end of the process, we observed 1396 samples as either implicit (no apparent hate span) or exhibiting hate intensity less than the threshold \( \tau \). Thus, our final gold dataset consists of 3027 parallel hateful and normalized samples, along with the intensity scores and hate spans in original ones (Table 2). Additionally, we annotated each normalized sample with its new hate intensity score. Two annotators were employed for this task.

Both annotators annotated all the available samples. In case of disagreement (11\% of cases), we employed a third annotator to break the tie. Overall the inter-annotator mean squared error for hate intensity was 0.20. In case of a disagreement on hate-span, if the non-overlap contains words like abuse or racial slurs, we added them to the span. In the rare case, if both annotators and third annotators were in disagreement over a span, the sample was dropped from the final dataset. Table 2 lists the overall statistics of the data curated and annotated. The detailed annotation guidelines are covered in Appendix C. Table 1 shows a few examples of the original hate samples, along with their normalized forms and intensity scores for both sets.\(^4\) A bar graph representing the original and normalized hate intensity distributions is shown in Figure 3. It is evident that the intensity distribution mass in original samples shifts towards the weaker intensity in normalized samples.

\textbf{Train and test sets}: We split the annotated dataset in the 70 : 15 : 15 for the training (2119), validation (454), and testing (455). One may rightfully argue that such a small number of samples in the test set (\( S_A \)) may not be adequate to evaluate the modules.

\begin{table}[h]
\centering
\caption{Dataset statistics.}
\begin{tabular}{|c|c|}
\hline
\textbf{Statistics} & \textbf{Value} \\
\hline
Total samples & 4423 \\
Sample length & 23 (avg), 112 (max) \\
No. of samples with intensity scores & 4423 \\
Hate intensity range & 1 (min) – 10 (max) \\
No. of samples with spans & 3027 \\
No. of spans & 5732 \\
Avg length of spans (tokens) & 5 \\
No. of normalized samples & 3027 \\
Normalized sample length & 21 (avg), 99 (max) \\
\hline
\end{tabular}
\end{table}

\(^3\)https://huggingface.co/

\(^4\)For annotations and human evaluation, the texts were not masked and presented as-us.
We design a three-way evaluation framework. First, we perform baselines that we deem pertinent to our work. It should be noted that we lack accurate baselines due to the novelty of the proposed task. Therefore, we adopt six existing methods as the first step toward extending the evaluation for adversarial attacks, which we hope to study in the future.

6 EVALUATION AND ANALYSES

We design a three-way evaluation framework. First, we perform an intrinsic evaluation to measure the model’s performance by traditional evaluation metrics. Second, we perform human evaluation where we ask human annotators to rate the quality of the generated output (from our model and the generation baselines). Third, we employ state-of-the-art hate speech detection models to discriminate between the original and normalized hate samples for extrinsic evaluation. Additionally, we evaluate NACL on datasets from three different platforms – Reddit, GAB, and Facebook. We further let human emulators supply their content in English and check in real-time the expected output for randomly supplied content. Both these external evaluations help test the robustness of the model against a host of inputs possible in the real world. This is also the first step toward extending the evaluation for adversarial attacks, which we hope to study in the future.

6.1 NACL-HIR Baseline Methods

It should be noted that we lack accurate baselines due to the novelty of the proposed task. Therefore, we adopt six existing methods as baselines that we deem pertinent to our work.

- **Dictionary-based Normalization** [44]: Inspired by the early work of Su et al. [44], we define a dictionary-based model that learns the mapping between the hateful span and its corresponding normalized span from the training set. We employ tf-idf based cosine similarity to perform a lookup to select the closest normalized span from the training set.

- **Neutralizing Subjective Bias** [40]: This style-transfer model aims to neutralize the subjective bias in news content by converting a piece of opinionated news into a neutral one. It utilizes an LSTM-based sequence-to-sequence model. We re-train the model for the hate normalization task, assuming the strongly hateful content is biased.

- **FGST** [33]: Given a sentence with a polarity label (positive or negative), FGST generates a new sentence having a pre-defined sentiment intensity score. In our setting, we aim to generate a normalized text having \( \phi_t \leq \tau \).

- **Style Transformer** [10]: Style Transformer is a transformer based architecture which generates a new sentence (for a given source sentence) without making any assumption about the latent representation of source sentence. We re-train the model for style transfer between original and normalized sentences.

- **Style Transfer (NPTCA)** [43]: Style Transfer (NPTCA) generates a new sentence for a chosen sentence by assuming that different corpora possess a shared latent content distribution. It uses two constrained versions of auto-encoders (aligned and cross-aligned) to refine the alignment of these latent space distributions. We re-train the model for style transfer between original and normalized sentences.

- **Style Transfer (DRG)** [32]: For a sentence, Style Transfer (DRG) generates a new sentence by modifying a specific attribute such as sentiment while keeping other attributes intact. It deletes phrases associated with the attribute in the original text and then retrieves new phrases associated with the target attribute and uses a sequence to sequence model to combine them. For our evaluations, we re-train the model for style transfer between original and normalized sentences.

6.2 Intrinsic Evaluation

We compute Pearson correlation, cosine similarity and RMSE scores for HIP (Table 3); whereas, we adopt F1-score for HSI (Table 4). In the case of HIR, which is a generative task, we compute perplexity and BLEU scores (Table 5). The model hyperparameters are covered in Appendix F. We iteratively build our modules, starting independently with the HIP and HSI modules and then using only the best-performing ones for HIR.

For HIP, as shown in Table 3, BERT+BiLSTM models performs better than others. Using BERT+BiLSTM with a linear activation gives the best results with cosine similarity of 0.973, Pearson score of 0.766 and an RMSE of 0.136. Meanwhile, BERT+LSTM with a sigmoid activation (scaling intensity between 0-1 range) reports a cosine similarity of 0.968, Pearson score of 0.704 and RMSE of 0.148. As overall the linear activation beats sigmoid, for the rest of our experimentation, we use BERT+LSTM (linear) as our HIP model.

For HSI, BiLSTM+CRF yields the best F1 of 0.622 and best Recall of 0.634 with ELMo embeddings as reported in Table 4. Among others, the fine-tuned SpanBERT model stands the second-best with 0.583 F1 and highest precision with 0.6913. For the rest of our experimentation, we use ELMo+BiLSTM+CRF as our HSI model.

Finally, we report the performance of NACL-HIR module along with other baselines in Table 5. HIP yields the best perplexity score.

![Figure 3: Hate intensity distribution for the original and normalized samples. The distribution for original samples shifts towards the weaker intensities in normalized samples.](image-url)

Table 3: Results: Hate intensity prediction (HIP).

| Model       | Description | Pearson ↑ | Cosine Sim ↑ | RMSE ↓  |
|-------------|-------------|------------|---------------|--------|
| CNN         | GLV         | 0.2827     | 0.2613        | 3.175  |
| BiLSTM      |             | 0.1939     | 0.2044        | 3.411  |
| BiLSTM+CNN  |             | 0.3060     | 0.3124        | 4.92   |
| CNN         | BERT        | 0.2549     | 0.212         | 2.9823 |
| BiLSTM      | BERT        | 0.2613     | 0.2457        | 4.9414 |
| BiLSTM+CNN  | BERT        | 0.3211     | 0.3375        | 3.691  |
| ReERT       | Fine-tune   | 0.9556     | 0.9527        | 0.7712 |
| ReERT       | ELMo        | 0.6521     | 0.6417        | 0.9682 |
| NACL-HIP    | BERT (linear)| 0.766     | 0.973         | 0.136  |
| NACL-HIP    | BERT (sigmoid)| 0.704     | 0.968         | 0.148  |
Table 4: Results: Hate span identification (HSI).

| Model          | Description | Evaluation Measure |
|----------------|-------------|---------------------|
|                |             | Precision ↑ | Recall ↑ | F1 Score ↑ |
| CRF            | GLV         | 0.7013       | 0.3867   | 0.4985     |
| Fine-tune      | BERT        | 0.6624       | 0.3537   | 0.4437     |
| NACL-HSI       |             | 0.6053       | 0.3676   | 0.4574     |
| SpanBERT       | Fine-tune   | 0.7081       | 0.5413   | 0.6135     |
|                | BERT        | 0.4917       | 0.4587   | 0.5470     |
|                | ELMO        | 0.6471       | 0.4823   | 0.5526     |
|                | SpanBERT   | 0.6913       | 0.5041   | 0.5830     |

(a lower value is better) of 80.05 for the generated normalized sentences. In comparison, the dictionary-based baseline obtains the perplexity of 92. For reference, we also compute perplexity (64.66) for the reference (gold) normalized sentences. Moreover, we observe a similar trend in the BLEU scores as well. The NACL-HIR model achieves the highest BLEU (82.27). The high BLEU score for HIR can be attributed to the fact that we mainly target hate spans for normalization, and a good portion of the original token (not containing hate) sequence gets preserved in the normalized sentence. Categorizing supervised and unsupervised based on whether the method requires parallel data, we observe that supervised methods like Bias Neutralization and some unsupervised methods like FGST perform moderately well in comparison to NACL-HIR. We observe abysmal performance from most unsupervised style transfer methods. This is mainly due to the lack of a large-scale corpus for training these models for our problem definition. Owing to which most of the generated sequences produce junk values. During the human evaluation, we do not consider these unsupervised baselines except for FGST.

6.3 Extrinsic Evaluation

Our hypothesis for extrinsic evaluation is that if a hate normalization model produces high quality normalized text with lower hate intensity score, then a hate speech detection method will exhibit less confidence in classifying the normalized text as hate. For this evaluation, we employ three widely-used hate speech detection methods – [47], [11], [19]. We train the methods on their respective evaluation, we employ three widely-used hate speech detection methods. This is mainly due to the lack of a large-scale corpus for training these models for our problem definition. Owing to which most of the generated sequences produce junk values. During the human evaluation, we do not consider these unsupervised baselines except for FGST.

Evaluation on Test Sets SA and SB: For each original sample t in \( S_A + S_B \), we extract \( \gamma(t, m) \), the softmax probability of the hate class as the confidence score, where m is the underlying hate detection method. Evidently, \( \gamma(\cdot) \in (0, 1) \). Subsequently, we compute the confidence score, \( \gamma(t', m) \) for each generated normalized sample \( t' \). Considering that hate speech normalization aims to reduce the hate intensity of samples instead of converting to non-hate, we restrict ourselves in analyzing a set of samples in \( M_m \subseteq S_A + S_B \) for which \( \gamma(t) \geq 0.5 \) and \( \gamma(t') \geq 0.5 \), i.e., both the original and normalized samples are predicted as hate. Finally, we compute the average difference in confidence score, \( \Delta \), for each pair \( (t, t') \) in \( M_m \) as,

\[
\Delta(t, t') = \frac{1}{|M_m|} \sum_{t \in M_m} \gamma(t) - \gamma(t') \tag{4}
\]

Table 6 reports the average difference in confidence scores for NACL-HIR and other baselines\(^5\). We observe the consistency of NACL-HIR over other normalization models as it obtains confidence scores \( > 0 \) across all three hate detection methods. In comparison, Style Transfer DRG gives the highest performance in two cases but reports a negative score in the third. It should be noted that all existing hate speech datasets are highly skewed against hateful labels [18]. Thus, detection models are prone to underperform for the hate class. Having considered that, extrinsic evaluation aimed at pointing out the general pattern in reduction of hatefulness post-normalization. Such behavior is expected to manifest irrespective of the hate detection model deployed for production.

6.4 Error Analysis

Though NACL performs better than other baselines across different evaluations, it does commit mistakes. Table 8 lists a few examples along with their original and predicted hate intensity scores. For the first example, NACL-HIP correctly predicts the hate intensity score as 10; however, for the other three examples, it wrongly predicts the score. NACL-HIP assigns lesser intensity score to the second utterance – the hatred is conveyed by a single word ‘qu**r’; and most likely, our model does not assign adequate focus on it in the presence of the positive phrase ‘happy birthday’. In the third example, the predicted high score is possibly because of the presence of three words ‘bi*ch’, ‘damn’, and ‘unfollow’. In comparison, we hypothesize that our model learns to associate color as a cue for hate, thus predicting a high hate intensity score for the last example.

We also encounter some misclassifications in hateful span identification, as shown in Table 10. In span identification, NACL-HIP commits three kinds of mistakes – (a) removes a token from hateful span during prediction; (b) includes additional tokens as part of the hateful span; and (c) classifies non-hate spans as hate. The first kind of mistake can be observed from examples 1 and 2 in Table 10.

\(^5\)Since we can’t generate the dictionary-driven normalized samples for test set in \( S_B \) in the absence of hateful spans (c.f. Section 6.1), we do not include it in our extrinsic evaluation.
Table 7: A few erroneous examples of NACL-HIR-generated vs. gold normalized texts.

| #  | Type         | Example                                                                 |
|----|--------------|-------------------------------------------------------------------------|
| 1. | Original     | LateNightThoughts how many Congressman [d***'s did women s**'k] to finaly gain voting rights |
|    | Reference    | LateNightThoughts how many Congressman [did women approach] to finally gain voting rights |
|    | Generated    | LateNightThoughts how many Congressman [did women s**'k] to finaly gain voting rights |
| 2. | Original     | [S**'s are half breed trash], No {filthy native} should be allowed to speak to any European. |
|    | Reference    | No [native] should be allowed to speak to any European. |
|    | Generated    | Mexicans are t**'k. No {disgusting native} should be allowed to speak to any European. |

while the second type is shown in example 3. Similarly, example 4 reflects the third type of mistake.

Finally, Table 7 lists a couple of examples considering the hate normalization task. In the first example, NACL-HIR partially normalizes the hate sample, and as a consequence, the generated text has low adequacy and fluency scores. Similarly, in the second, we see: (a) the generated sentence is relatively more fluent than the first; and (b) the intensity value is on the higher side due to the presence of phrases ‘Mexicans are t**k’ and ‘disgusting’. We argue that the problem of partial normalization can be effectively addressed with more volume and variety of training samples.

6.5 Human Evaluation

In order to check the viability of the overall system, aided by Wipro AI, we perform a human evaluation and assess the generated text from NACL-HIR and other baselines. We prepare a questionnaire to measure the qualitativeness of the generated texts. Evaluation instructions are listed in Appendix E. For a subset of hateful samples from our dataset, the human annotators are provided with four outputs corresponding to four high performing hate normalization systems, i.e., Dictionary-based, Bias Neutralization, FGST, and NACL. To reduce bias, we anonymize the systems, randomly shuffled the outputs, and labeled them as A, B, C, D. Given an original sample, we ask 20 human annotators to evaluate the generated texts on three dimensions – hate intensity, adequacy, and fluency [4]. Adequacy (the higher, the better) measures the semantic perseverance in the generated text, while fluency (the higher, the better) refers to the linguistic smoothness in the target language. We modify the definition of adequacy to adopt it for the hate speech normalization task. A predicted sentence that does not reduce hate intensity is considered inadequate and should have a lower adequacy score. We also provide the intensity of the original sample for reference. Finally, we aggregate the average intensity, adequacy, and fluency scores, as shown in Table 9. We present the average scores for one sample and the overall average scores across all samples. On average, NACL-HIR outperforms others, with hate intensity of 4.08, fluency of 4.38, and adequacy of 4.16.

6.6 Platform-Independence Analysis

To further check the robustness of our tool, we extend the human evaluation to a cross-platform analysis of NACL. We evaluate on randomly selected 100 samples from Reddit, GAB and Facebook hateful posts, obtained from [41] (for first two) and [9]. We employ the same set of annotators and annotation process mentioned in Section 6.5 to evaluate the quality of NACL and two best baselines in terms of intensity, adequacy, and fluency. Table 11 shows that NACL performs convincingly well compared to others across platforms.

7 RELATED WORK

Hate Speech Detection: From the simple logistic regression to [11, 47] deep learning-based models [2, 30], the work on hate speech has diversified in volume [6, 18], languages [36], granularity and variety[1]. Fortuna and Nunes [17], and Poletto et al. [39] put together a detailed survey of various hate speech detection methods and their shortcomings. In this work, we do not propose any new hate speech detection method. Our framework comes into use once a speech is detected as hateful. Thus, hate speech detection underpins the work of hate speech normalization.

Rephrasing Hateful Text: In the context of rephrasing offensive text, Su et al. [44] led the initial work by building a rule-based system (29 hand-crafted rules) for rephrasing profane Chinese phrases/terms. We, too, tested such a tf-idf based mapping approach, but the limitations of such rule-based systems are in terms of out of vocabulary (OOV) phrases. In another work, Nogueira dos Santos et al.

Table 8: Gold and predicted values for hate intensity prediction using our proposed model.

| #  | Example                        | Gold | Pred |
|----|--------------------------------|------|------|
| 1. | b***'h you're a grown a** women @user | 10   | 10   |
| 2. | @user happy birthday qu**r       | 8    | 4    |
| 3. | @user @user @user Man b***'h am not gonna do a damn thing , if u don't like what I got to say then unfollow me | 4    | 10   |
| 4. | firefighter is a job for white people | 2    | 8    |

6. Among 20 annotators, 10 were male, and 10 were female. The age of all the annotators ranged between 25-40 years. All of them were social media savvy.

8https://palletsprojects.com/p/flask/

6 A demo video of our tool https://github.com/LCS2-IIITD/Hate_Norm
Table 9: The left part of the table shows an example of human evaluation where we give the generated text to human annotators to evaluate the output of each competing model on three parameters – intensity, fluency, and adequacy. The results are an average from all the responses received for this example. The right part of the table shows the average scores of all three parameters across all responses (i.e., overall system averages). Intensity ranges from 1 – 10, Fluency and Adequacy from 1 – 5.

| Model       | Tweet                                    | Int ↓ | Flu ↑ | Adq ↑ | Int ↓ | Flu ↑ | Adq ↑ | Overall Average  |
|-------------|------------------------------------------|-------|-------|-------|-------|-------|-------|------------------|
| Original    | @user you shouldn’t be putting it out there c** n** et | 6     | -     | -     | -     | -     | -     | 4.13             |
| NACL        | @user you shouldn’t be putting it out | 5.72  | 3.45  | 3.45  | 3.45  | 3.45  | 3.45  | 3.61             |
| Bias Neur   | @user you shouldn’t be putting it out | 4.72  | 3.82  | 3.82  | 3.82  | 3.82  | 3.82  | 3.61             |
| FGST        | you shouldn’t be putting it there woman n** et | 3.28  | 2.45  | 2.64  | 2.64  | 2.64  | 2.64  | 3.05             |
| FGST        | you shouldn’t be putting it there woman n** et | 2.36  | 4.54  | 3.90  | 3.90  | 3.90  | 3.90  | 3.61             |

Table 10: Gold and predicted spans for HSI, with spans indicated by blue text. The green text in the second column indicates misprediction – either new tokens being inserted into the span or gold tokens being excluded from the predicted span.

| #          | Gold Span                  | Predicted Span                  |
|------------|----------------------------|----------------------------------|
| 1.         | Why do girls follow me ? All I do is make fun of them [for how dumb they are] | Why do girls follow me ? All I do is make fun of them [for how dumb they are] |
| 2.         | [go f** yourself you stupid ugly c** ] @user | go f** yourself you stupid ugly c** @user |
| 3.         | Blac Chyna is [straight trash and an abomination to women] | Blac Chyna is [straight trash and an abomination to women] |
| 4.         | okay fl** h, fl** k off its not your business [fall in a hole and get a** f** d by satans horn] | okay fl** h, fl** k off its not your business [fall in a hole and get a** f** d by satans horn] |

Table 11: Human evaluation on three other platforms.

|       | Reddit [41] | GAB [41] | Facebook [9] |
|-------|-------------|----------|--------------|
|       | Int ↓ | Flu ↑ | Adq ↑ | Int ↓ | Flu ↑ | Adq ↑ | Int ↓ | Flu ↑ | Adq ↑  |
| FGST  | 5.12  | 2.26  | 1.63  | 5.70  | 2.32  | 1.43  | 6.08  | 2.59  | 1.45  |
| Bias  | 3.28  | 2.02  | 1.01  | 3.89  | 1.82  | 1.02  | 4.67  | 2.41  | 1.06  |
| NACL  | 3.25  | 3.8   | 1.92  | 3.29  | 4.25  | 2.71  | 3.2   | 4.05  | 2.6   |

[37] built an unsupervised text-transfer model for tackling offensive language across social media platforms. In the latest work on unsupervised text transfer for profanity, Tran et al. [46] employed a vocabulary-based approach to determine if a sentence should be considered as profane and then proceed to generate its non-offensive version. The works by both Tran et al. [46] and Nogueira dos Santos et al. [37] are similar to our use case. However, they differ in that they employed an unsupervised approach that depended on classifiers or lexicons to capture the offensive/profane text and also aim at 180° transformation into a non-offensive one. In contrast, our aim is the reduction of hate and not the complete absence of it. On the line of negativity reduction, Madaan et al. [34] employed a tag-based approach for increasing the politeness of a question/answer (query) pair. Pryzant et al. [40] adopted the similar tag-based approach but for opinionated news sentences, which is closer to our task than a Q&A-based setup. Starting with Xu et al. [48], several studies (both supervised and unsupervised) showed successful rephrasing of a sentence by modifying its sentiment attribute. These methods [10, 22, 32, 42, 43] largely disentangled the sentiment attribute and then relied on a combination of select rephrase and attention mechanism to generate an output sequence. For our use case, we hope to learn the hateful-span attributes. Our experiments show that alternatively reducing bias or sentiment is inadequate for reducing hate intensity due to the subjective nature of hate speech. Additionally, we observed that unsupervised style transfer methods that rely on sizeable monolithic corpus hardly perform well when trained on low-volume datasets such as ours.

8 CONCLUSION AND FUTURE SCOPES

To combat the severity of online hate speech, we proposed an alternative solution by introducing the novel task of hate speech normalization. To this end, we proposed NACL, a neural hate speech normalizer. We collected and manually annotated a new dataset to support the task. We performed exhaustive evaluations to establish the model’s efficacy under different settings.

We observed two major challenges for the hate normalization task – first, the lack of parallel data to train more sophisticated generative models, and second, the presence of implicit hate in samples. Though the first hurdle can be addressed (albeit expensive) by annotating more samples, handling the implicit hate is cumbersome [14].

In the current work, we skipped over the implicit hateful samples due to the absence of explicit hate spans. In the future, we would like to put in rigorous effort to handle such cases and increase the size of the dataset. Additionally, it would be interesting to see how NACL can be extended to non-English texts.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the Prime Minister Doctoral Fellowship (SERB India), Ramanujan Fellowship (SERB, India), Infosys Centre for AI (CAI) at IIIT-Delhi, and ihub-Anubhuti-iitiid Foundation set up under the NM-ICPS scheme of the DST, India. We would also like to thank our industry partner Wipro AI. Wipro is an Indian multinational conglomerate with diverse businesses, coordinated the field study for possible deployment. We acknowledge the support of Shivam Sharma, Technical Lead, Wipro AI for the same. We thank all the human subjects for their help in evaluating our tool.
REFERENCES

[1] Md Rabiiul Awal, Rui Cao, Roy Ka-Wei Lee, and Sandra Mitrovic. 2021. Angry-BERT: Joint Learning Target and Emotion for Hate Speech Detection. In PAKDD. 74–85.

[2] Pinkesh Badjatia, Shashank Gupta, Manish Gupta, and Vasudev Varma. 2017. Deep Learning for Hate Speech Detection in Tweets. In WWCW. 759–760.

[3] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Deborah Nozza, Vivianna Patti, Francesco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In Proceedings of the 13th International Workshop on Semantic Evaluation. 54–63.

[4] Pushpaj Bhattacharyya. 2015. Machine Translation. Chapman and Hall/CRC. 260 pages.

[5] Md Momen Bhuiyan, Kexin Zhang, Michael A. Horning, and Pushpak Bhattacharyya. 2015. A Simple Approach to Sentiment and Style Transfer. In Findings of ACL-IJCNLP 2015. Online, 59–69.

[6] Rui Cao and Roy Ka-Wei Lee. 2020. HateGAN: Adversarial Generative-Based Data Augmentation for Hate Speech Detection. In Coling. 6327–6338.

[7] Mudit Chaudhary, Chandni Saxena, and Helen Meng. 2021. Countering Online Hate Speech: An NLP Perspective. arXiv:2106.02941 [cs.CL].

[8] Justin Cheng, Lada Adamic, P. Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. 2014. Can Cascades Be Predicted?. In WWW 925–936.

[9] Yi-Ling Chung, Elizaveta Kurmzenko, Serra Sinem Tekرؤgرو، and Marco Guerini. 2019. CONAN - CoNtrAattive via Luxichosourcing: A Multilingual Dataset of Responses to Fight Online Hate Speech. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, 2819–2829. https://doi.org/10.18653/v1/P19-1271.

[10] Ding Dai, Jianze Liang, Xinpeng Qu, and Xuanjing Huang. 2019. Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation. In ACL. 5997–6007.

[11] Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial Bias in Hate Speech and Abusive Language Detection Tasks. In Proceedings of the Third Workshop on Abusive Language Online. Florence, Italy. 25–35.

[12] Ona de Giber, Naiara Perez, Tom Mas-Caballero, and Montse Pan. 2018. Hate Speech Dataset from a White Supremacy Forum. In Proceedings of the 37th International Conference on Machine Learning - Volume 70 (Sydney, NSW, Australia) (ICML’17). JMLR.org. 1587–1596.

[13] Andrew Hutchins. 2019. Instagram Adds New Anti-Bullying Measures, Including Comments, Story, and User Restrictions. https://bit.ly/30easaj. Accessed: 2020-10-30.

[14] Andrew Hutchins. 2020. Twitter Is Testing Prompts Which Would Recommend Users Hide Potentially Offensive Replies. https://bit.ly/3F3EeqW. Accessed: 2020-10-30.

[15] Akshita Jha and Radhika Mamidi. 2017. When does a compliment become sexist? Analysis and classification of ambivalent sexism using twitter data. In Proceedings of the Second Workshop on NLP and Computational Social Science. Vancouver, Canada, 7–16.

[16] Di Jin, Zheng Jin, Zhixing Hu, Olga Vetchonova, and Rada Mihalcea. 2020. Deep Learning for Text Style Transfer: A Survey.

[17] N. F. Johnson, R. Leahy, N. JohnsonRestrepo, N. Velasquez, M. Zheng, P. Manrique, P. Devkota, and S. Wuchty. 2019. Hidden resilience and adaptive dynamics of the global online hate ecology. Nature 573, 7773 (01 Sep 2019), 261–265.

[18] Matthew Katsaros, Kathy Yang, and Lauren Fratamico. 2021. Reconsidering Tweets: Intervening During Tweet Creation Decreases Offensive Content. arXiv preprint arXiv:2112.00773 (2021).

[19] Geoff F Kaufman and Lisa K Libby. 2012. Changing beliefs and behavior through experience-taking. Journal of personality and social psychology 103, 1 (July 2012), 1–19. https://doi.org/10.1037/a0027525

[20] Douwes Kjela and Lisa K Libby. 2012. Changing beliefs and behavior through experience-taking. Journal of personality and social psychology 103, 1 (July 2012), 1–19. https://doi.org/10.1037/a0027525

[21] Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial Bias in Hate Speech and Abusive Language Detection Tasks. In Proceedings of the Third Workshop on Abusive Language Online. Florence, Italy. 25–35.

[22] Paula Fortuna and Sérgio Nunes. 2018. A Survey on Automatic Detection of Hate Speech in Text. ACM Comput. Surv. 51, 4 (2018).

[23] Antigoni Founta, Constantinos D jovouas, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Gianluca Strighini, Athina Vakali, Michael Svirinovas, and Nicolas Kourtellis. 2018. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. In ICWSM. 491–500.

[24] Antigoni Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilia Leontiadis. 2019. A Unified Deep Learning Architecture for Abuse Detection. In WebSci. 105–114.

[25] Lei Gao and Ruishuang Hong. 2017. Detecting Online Hate Speech Using Context Aware Models. In RANLP. 260–266.

[26] Dominik Hangartner, Gloria Gennaro, Sary Alasri, Nicholas Babich, Alexander Borborth, Joseph Boucher, Buke Buset Demirci, Laurens Derksen, Aldo Hall, Matthias Jochum, Maria Murias Munoz, Marc Richter, Franziska Vogel, Salome Wittwer, Felix Wüthrich, Fabrizio Gilardi, and Karsten Donnay. 2021. Empathy-based counterspeech can reduce racist hate speech in a social media field experiment. PNAS 118, 50 (2021). https://doi.org/10.1073/pnas.2106310118

[27] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2019. Toward Controlled Generation of Text. In Proceedings of the 34th International Conference on Machine Learning - Volume 70 (Sydney, NSW, Australia) (ICML’17). JMLR.org. 1587–1596.

[28] Andrew Hutchins. 2019. Instagram Adds New Anti-Bullying Measures, Including Comments, Story, and User Restrictions. https://bit.ly/30easaj. Accessed: 2020-10-30.

[29] Andrew Hutchins. 2020. Twitter Is Testing Prompts Which Would Recommend Users Hide Potentially Offensive Replies. https://bit.ly/3F3EeqW. Accessed: 2020-10-30.

[30] Akshita Jha and Radhika Mamidi. 2017. When does a compliment become sexist? Analysis and classification of ambivalent sexism using twitter data. In Proceedings of the Second Workshop on NLP and Computational Social Science. Vancouver, Canada, 7–16.
APPENDIX

A ETHICAL IMPLICATIONS

As pointed out by [26], style transfer techniques in NLP can be a force for both good and bad. While, on the one hand, they help turn the Internet into a more positive space [37], they can also be used to propagate a false/malicious narrative. Reiterating that, our research aims not to support the hateful users but rather to study an alternative approach to prompt users towards a less toxic enunciation of their opinions. Without forcing the users to jump from a negative to a positive space directly, we operate on the space in between as a stepping stone and study the theoretical effectiveness of such a technique (the practical effect is in parts highlighted by human evaluation in Section 6.5). Keeping in mind the nefarious use of the normalized hate speech (without the knowledge of online platforms about such a normalization), we will not be publicly releasing the parallel corpus. However, it shall be made available to researchers subject to their request for fair usage.

B FEATURES FOR VIRALITY PREDICTION

The text-based features employed for predicting comment engagement (aka virality) of the data samples are listed below:

- Complexity: Uniqueness of terms introduced as measured by the logarithm of term-frequency in the test samples.
- Readability: LIX and RIX readability scores.\(^9\)
- Informativeness: Summation of the TF-idf vectors of the words in the sample.
- Polarity: Overall SentiNet\(^10\) polarity score of the test sample.

C DATASET ANNOTATION GUIDELINE

For this experiment, we followed the definition proposed by [47] for hate speech and marked the hate span if it consists of any of the following explicit mentions:

- A sexist or racist slur term, or an abusive term directly attacking a minority group/individual.
- A phrase that advocated violent action or hate crime against a group/individual.
- Negatively stereotyping a group/individual with unfounded claims or false criminal accusations.
- Hashtag(s) supporting one or more of the points as mentioned earlier.

Additionally, the hate intensity of a sample was marked on a scale of 1 – 10, 10 being the highest based on:

- Score[8 – 10]: The sample promotes hate crime and calls for violence against the individual/group.
- Score[6 – 7]: The sample is mainly composed of sexist/racist terms or portrays a sense of gender/racial superiority on the part of the person sharing the sample.
- Score[4 – 5]: Mainly consists of offensive hashtags, or most hateful phrases are in the form of offensive hashtags.
- Score[1 – 3]: The sample uses dark humor or implicit hateful term.

D ALGORITHM

Algorithm 1 shows the learning protocol of NAACL.

Algorithm 1 Learning NAACL

```
Input: A set of hateful samples \( T = \{t_1, ..., t_m\}; \forall j \quad \phi_j \geq \tau \)
Output: A set of normalized samples \( T' = \{t'_1, ..., t'_m\}; \forall j \quad \phi'_j < \tau \)

procedure NAACL(T, \( \tau \))
Pre-train HSI(T) and HIP(T)
repeat
  for all \( t \in T \) do
    \([s_1, e_1], ..., (s_k, e_k)\) \(\rightarrow\) HSI\((t)\) \(\rightarrow\) Get indices of hate spans.
    \(t' \leftarrow HIR(t, s_1, e_1, ..., (s_k, e_k))\) \(\rightarrow\) Generator: Get normalized text.
    \(\phi' \leftarrow HIP(t')\) \(\rightarrow\) Discriminator: Get the intensity of the normalized text.
    \(R = \tau - \phi'\) \(\rightarrow\) Calculate reward for the generator - \(\pm\)ve, if intensity less than threshold, -\(\pm\)ve, otherwise.
    \(L = t + (1 - R)\) \(\rightarrow\) \(t \rightarrow\) Generator loss
    HIR \(\rightarrow\) Backpropagation(L)
  end for
until \langle termination-condition \rangle
return \(T'\)
```

procedure HIP(t)

```
\(H' \leftarrow \text{BiLSTM}(\text{BiLSTM}(t))\)
\(\alpha \leftarrow \text{Attention}(H')\)
\(H = H' \cdot \alpha\)
\(\phi_l \leftarrow \text{Linear}(H)\)
return \(\phi_l\) \(\rightarrow\) Hate intensity of the tweet.
```

procedure HSI(t)

```
\(h_1, ..., h_n \leftarrow \text{BiLSTM}(w_1, ..., w_n)\)
\(p_1, ..., p_n \leftarrow \text{CRF}(h_1, ..., h_n)\) \(\rightarrow\) \(\forall i \quad p_i \in \{B, I, O\}\)
\([s_1, e_1], ..., (s_k, e_k)\) \(\rightarrow\) GetSpans(p_1, ..., p_n) \(\rightarrow\) \(\langle s, e \rangle\) are the start and end indices of the hate span.
return \([s_1, e_1], ..., (s_k, e_k)\)
```

procedure HIR(t, \(s_1, e_1, ..., (s_k, e_k)\))

```
t' \leftarrow t
for all \(\langle s_i, e_i \rangle\) do
  \(\rightarrow\) For each identified hate span.
  \(H \leftarrow \text{BART-encoder}(t)\)
  \(t'_i[s_i, e_i] \leftarrow \text{BART-decoder}(H, s_i, e_i)\) \(\rightarrow\) Update hate span with normalized text.
end for
return \(t'\)
```

E HUMAN EVALUATION GUIDELINE

After reading the original sample and the normalized counterparts, the annotators provided their input in the form of the following:

- Intensity: The annotators assigned a hateful intensity score to each generated output on a scale of \([1,10]\), 10 being the highest intensity.
- Fluency: To understand how well constructed and readable the generated text is, the annotators scored each generated text on a range of \([1,5]\), 5 being the highest fluency.
- Adequacy: Additionally, to provide an idea of whether the desired meaning can be interpreted from the output, the annotators were asked to score each generated text on a range of \([1,5]\), 5 being
Figure 4: Snapshots of the web extension for four scenarios. NACL generates normalized text only if $\phi_t > \tau$. The web framework detects hate as the user types in, and if any $\phi_t > \tau$, it shows the level of hate that is detected in the current text, and then recommends the normalized text to the user.

the highest. Since our task aims to perform normalization and not the conversion of hate to non-hate, if a sentence changes the sample’s polarity, then that can also be taken as a negative case from our intended perspective. The annotators were informed before that a normalized sample with its polarity reversed would have a minimum (1) adequacy even if it is fluent.

For reference, the intensity of the original sample was provided. The fluency and adequacy of the original sample were considered highest. To reduce the annotator’s bias, the annotators were unaware of which output text represented our system.

F HYPERPARAMETERS
The models make use of Tensorflow 2.0 and Transformer 4.5.1 with Python 3 libraries, trained on Google Colab with Tesla P100-PCIE-16GB GPU.

- For the HIP model, we employ a single Bi-LSTM layer ($hdim=512$) followed by a layer of self-attention. For this model, use the MSE loss and the Adam optimizer with linear activation. The input embedding is BERT $dim=768$. The model is trained for 10 epochs with a batch size of 32.
- For the HSI, we employ a 2 layer Bi-LSTM ($hdim=512$), using a batch size of 32, and the RELU activation function. As the final layer for the Span Model is a CRF, we take crf_loss and crf_accuracy as our loss and accuracy metric respectively, and Adam optimizer. The model is trained for 5 epochs with a batch size of 32.
- For the HIR, we employ Facebook’s BART-base module, with a cross-entropy loss, RELU activation and Adam optimizer.

G DEPLOYMENT DETAILS
The proposed web-interface can easily be made to work in cross platform settings to proactively curb hate speech, and the screenshot for the same is provided in Figure 4.

Again we evaluate the tool in the wild by asking 25 participants to write random hateful content on their own and rate the tool’s output. We extend the tool by including the interface of the Bias Neutralization baseline – the tool shows the outputs of NACL and the baseline for a given input. Each participant is asked to input hateful content, assign an original intensity score, and subsequently evaluate the tools’ outputs considering fluency, adequacy, and intensity. The normalization methods are anonymized for the participants. In total, we obtain 100 input samples, with an average normalized intensity score of 3.24. Similar to earlier observations, we notice that NACL results in more fluent sentences with higher reduction in intensity. This in-the-wild evaluation further supplements that NACL is not restricted to our dataset.

H EXTRINSIC EVALUATION
The distribution of hateful and non-hateful samples for the three hate speech detection models we use in our extrinsic evaluation.

| Table 12: The hate vs non-hate tweet ratio for the hate classification. The data is collected from the respective methods. |
|----------------|----------------|----------------|
| Model          | # of tweets   | Hate:Non-Hate |
| Waseem and Hovy [47] | 12772          | 4750:8022      |
| Davidson et al. [11] | 24783          | 20620:4183     |
| Founta et al. [19] | 59189          | 13551:45638    |