Generalizable Implicit Hate Speech Detection using Contrastive Learning

Youngwook Kim\textsuperscript{1}, Shinwoo Park\textsuperscript{2} and Yo-Sub Han\textsuperscript{1}
\textsuperscript{1}Department of Computer Science, Yonsei University, Seoul, Republic of Korea
\textsuperscript{2}Department of Artificial Intelligence, Yonsei University, Seoul, Republic of Korea
\{youngwook, pshkhh, emmous\}@yonsei.ac.kr

Abstract

Hate speech detection has gained increasing attention with the growing prevalence of hateful contents. When a text contains an obvious hate word or expression, it is fairly easy to detect it. However, it is challenging to identify implicit hate speech in nuance or context when there are insufficient lexical cues. Recently, there are several attempts to detect implicit hate speech leveraging pre-trained language models such as BERT and HateBERT. Fine-tuning on an implicit hate speech dataset shows satisfactory performance when evaluated on the test set of the dataset used for training. However, we empirically confirm that the performance drops at least 12.5\% in F1 score when tested on the dataset that is different from the one used for training. We tackle this cross-dataset underperforming problem using contrastive learning. Based on our observation of common underlying implications in various forms of hate posts, we propose a novel contrastive learning method, \textit{ImpCon}, that pulls an implication and its corresponding posts close in representation space. We evaluate the effectiveness of \textit{ImpCon} by running cross-dataset evaluation on three implicit hate speech benchmarks. The experimental results on cross-dataset show that \textit{ImpCon} improves at most 9.10\% on BERT, and 8.71\% on HateBERT.

1 Introduction

\textbf{Warning:} this paper contains contents that may be offensive or upsetting.

Hate speech is “any communication that disparages a target group of people based on some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic” (Nockleby, 2000). Recently, there are several attempts to detect hate speech or abusive text using lexicon-based methods (Chen et al., 2012; Gitari et al., 2015; Lee et al., 2018; Wiegand et al., 2018) or neural-based methods (Gambück and Sikdar, 2017; Badjatiya et al., 2017; Park and Fung, 2017; Zhang et al., 2018; Lee et al., 2019; Wang et al., 2020). While these approaches work fairly well when a text contains an explicit hate or abusive word, they often fail to detect implicit ones. See Table 1 for examples of implicit hate speech. Caselli et al. (2020) showed that the pre-trained language model struggles to detect implicit abusive- ness. They suspect that a small amount of training data with implicit abusiveness is a main reason for the poor performance. In this vein, ElSherief et al.

<table>
<thead>
<tr>
<th>Text (Input) :</th>
<th>“my world orbits around whites as it should . laughable moment though .”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label :</td>
<td>Hate</td>
</tr>
<tr>
<td>Text (Input) :</td>
<td>“that is part of the white supremacy logic that native people are less than human . we aren’t .”</td>
</tr>
<tr>
<td>Label :</td>
<td>Not Hate</td>
</tr>
<tr>
<td>Text (Input) :</td>
<td>“send them back to the countries they came from”</td>
</tr>
<tr>
<td>Label :</td>
<td>Hate</td>
</tr>
</tbody>
</table>

Table 1: Example input texts and labels (Hate / Not Hate) from IMPLICIT HATE CORPUS (IHC) (ElSherief et al., 2021) which is an implicit hate speech dataset.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Implicit hate speeches and their shared implication from IMPLICIT HATE CORPUS (IHC).}
\end{figure}
(2021) recently presented an implicit hate speech benchmark. The models trained on this dataset outperform other baselines in terms of in-dataset evaluation performance. In general, the hate speech detection performance can be over-estimated when evaluated on its own test set (Arango et al., 2019; Yin and Zubiaga, 2021). On in-dataset evaluation, a model is evaluated on the test set of the same dataset used for training. However, on cross-dataset evaluation, a model is evaluated on the dataset that is different from the one used for training. Instead of in-dataset evaluation, it is better to run a cross-dataset evaluation to see the generalization ability of a model (Wiegand et al., 2019). As a preliminary experiment, we perform the cross-dataset evaluation for the current state-of-the-art models trained on implicit hate speech datasets. In Section 2.2, we empirically observe relatively low performance on cross-dataset evaluation.

Prior research (Gunel et al., 2021) incorporates contrastive learning into their fine-tuning process, resulting in better generalization ability in few-shot learning setup. Motivated by this, we propose contrastive learning methods to improve the generalization ability of implicit hate detectors on cross-dataset. Contrastive learning makes positive pairs to be close together and negative pairs to be apart in the representation space (Rethmeier and Augenstein, 2022). One of the key issues in contrastive learning is how to choose positive samples. Depending on different choices of positive sampling, a model would learn different invariant features (Tian et al., 2020). Here, we suggest two positive sampling strategies: 1) Leveraging augmented posts as positive samples of given posts (AugCon); 2) Leveraging implications as positive samples of given hateful posts (ImpCon). For AugCon, we first generate augmented posts which are lexically different but semantically similar with their original posts. ImpCon leverages implications as positive samples of hateful posts, since it contains concealed meaning of the hateful posts. In addition, a common implication is often shared by a group of hateful posts, as shown in Figure 1. By pulling an implication and its corresponding hateful posts close in representation space, the model can learn common features among a group of hateful posts sharing an implication.

We evaluate the generalization ability of models trained using AugCon and ImpCon. We conduct cross-dataset evaluation on three implicit hate speech benchmarks with BERT and HateBERT as base models. By incorporating AugCon or ImpCon in fine-tuning, we can improve the cross-dataset evaluation performance. While improvement with AugCon is limited to BERT (at most 2.92% improvement), ImpCon brings consistent improvements across all cross-datasets and models (at most 9.10% improvement to BERT and 8.71% improvement to HateBERT). The consistent improvement of ImpCon demonstrates the effectiveness of leveraging implication-post pair on the generalization ability. Moreover, further analysis on ImpCon shows that even unseen implication-post pairs are projected closer on the representation space (Section 5.1), resulting in consistent predictions on cross-dataset (Section 5.2). Our code is available at https://github.com/youngwook06/ImpCon.

2 Related Work and Preliminary Experiment

2.1 Hate Speech Detection

With the increase of online media and user contents, hate speech becomes more pervasive online. Considering the massive volume of online posts, it is impractical to manually moderate all posts. Researchers have developed many hate speech detection models, including lexicon-based approaches (Chen et al., 2012; Gitari et al., 2015; Lee et al., 2018; Wiegand et al., 2018) and neural network models (Gambäck and Sikdar, 2017; Badjatiya et al., 2017; Park and Fung, 2018).
2017; Zhang et al., 2018; Lee et al., 2019; Wang et al., 2020). Also, there are several datasets available for hate speech detection with different focuses (Warner and Hirschberg, 2012; Davidson et al., 2017; Founta et al., 2018; Basile et al., 2019). For example, Davidson et al. (2017) introduced a dataset to distinguish hate speech from an offensive language and Founta et al. (2018) investigated representative labels by merging and eliminating some labels related to abusive tweets. However, many of these datasets are skewed towards explicit forms of abusiveness since the data collection strategies often rely on explicit signals such as hateful lexicons (ElSherief et al., 2021). A model trained on such dataset often fails to detect implicit hate, even for the pre-trained language model (Caselli et al., 2020).

Recently, researchers show their interests in addressing implicit hate or abusiveness. Han and Tsvetkov (2020) used a set of probing data for the robust classifier which better detects disguised toxicity. Wiegand et al. (2021) studied subtypes of implicit abuse and existing datasets. ElSherief et al. (2021) presented a benchmark with implicit hate label, annotated target and implication.

2.2 Preliminary Experiment

Several works in hate speech detection have reported a large drop of the fine-tuned model performance when evaluated on cross-dataset (Gröndahl et al., 2018; Arango et al., 2019; Swamy et al., 2019). We conduct a preliminary experiment to see if implicit hate speech detection models can still perform well on cross-datasets that are also skewed towards implicit hate. We use three implicit hate datasets (IMPLICIT HATE CORPUS (IHC), SOCIAL BIAS INFERENC CORPUS (SBIC) and DYNABATE) following Hartvigsen et al. (2022). Detailed descriptions of the datasets are presented in Section 4.1. We experiment with one of the state-of-the-art models, BERT (Devlin et al., 2019). We also experiment with HateBERT (Caselli et al., 2021), which is pre-trained on abusive corpus and showed better generalization ability than BERT in their paper. In the cross-dataset evaluation with implicit hate datasets, we observe the similar generalization issue. As shown in Table 2, the performance of both models drops consistently over 12.5% in F1 score across implicit hate speech datasets. Through the preliminary experiment, we conclude that implicitly trained models suffer from generalization issue and combating the issue is needed.

2.3 Contrastive Learning

Recently, contrastive learning has been widely used to learn representation in various domains and showed its effectiveness. Many works on contrastive learning have proposed diverse choices of positive sampling.

For example, in the computer vision field, SimCLR (Chen et al., 2020) applies random augmentation on images and those augmented images from the same image are considered positive. Khosla et al. (2020) proposed to use the samples from the same class for positive sampling. In the natural language processing field, CERT (Fang et al., 2020) augments text with back-translation and considers augmented texts from the same text as positive. Also, Giorgi et al. (2021) suggested leveraging textual segments nearby in the document as positive samples. Gao et al. (2021) proposed using pairs from natural language inference datasets for positive sampling.

Some works on text classification proposed to apply contrastive learning to fine-tune the model. Gunel et al. (2021) showed that pulling instances from the same class closer while fine-tuning improved few-shot learning performance. Suresh and Ong (2021) extended this approach and showed that weighting negative samples differently increased performance on fine-grained classification. Pan et al. (2022) used adversarial examples as positives and showed outperforming performance over standard fine-tuning. We suggest using contrastive learning in the fine-tuning process for generalizable implicit hate speech detection.

3 Approach

3.1 Overall Training Objective

Generally, hate speech detection models are fine-tuned in a supervised way using the following cross-entropy loss $L_{ce}$:

$$L_{ce} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)], \quad (1)$$

where $N$ is the number of input posts in a batch, $\hat{y}_i$ indicates the model predicted probability of $i$-th input $x_i$ and $y_i$ is the ground-truth label of $x_i$, respectively. However, since cross-entropy loss has limitation on making large inter-class margin or intra-class compactness, fine-tuning using only
my world orbits around whites as it should. laughable moment though.

Encoder
Shared
weights
People of color are inferior to white people

Imp(x_i)

Figure 2: Left: The overview of training a model with \( \mathcal{L}_{imp} \) (Equation (7)). We present a hateful post and its implication as an example positive pair for \( \mathcal{L}_{imp} \) (Equation (6)). Right: ImpCon aims at pulling an implication and its corresponding hateful posts, resulting in similar representation between a shared implication and its hateful posts in the representation space. All implication and hateful posts in the figure are from IHC dataset.

cross-entropy loss can result in suboptimal generalization (Liu et al., 2016; Zhao et al., 2021).

We propose to combine contrastive loss with cross-entropy loss to train generalizable implicit hate speech detector. Contrastive loss pushes the representation of positive pairs closer and negative pairs further apart. We denote the positive sample of \( x_i \) as \( x_i^{pos} (i \geq 1) \). Given \( N \) training input posts in a batch, we assume one positive sample per post, leading to total \( 2N \) samples in a batch. When \( x_i^{pos} \) is the \( j \)-th input in a batch, i.e., \( x_i^{pos} = x_j \), we assume \( j = i + N \) if \( i \leq N \) and \( j = i - N \) if \( i > N \). We consider all samples other than a positive sample as negative samples, excluding itself. Following Chen et al. (2020), the contrastive learning loss \( \mathcal{L}_{cl} \) can be defined as:

\[
\mathcal{L}_{cl} = -\frac{2N}{i=1} \log \frac{e^{h(x_i) \cdot h(x_i^{pos})/\tau}}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} e^{h(x_i) \cdot h(x_k)/\tau}}, \tag{2}
\]

where \( \cdot \) denotes dot product operation, \( h(x_i) \in \mathbb{R}^H \) is the representation of the encoder for input \( x_i \), and \( H \) is the hidden dimension size. In detail, the last layer representation of [CLS] token is further normalized and used as \( h(x_i) \) for input \( x_i \). \( \mathbb{1}_{[i]} \) is an indicator function and \( \tau \) is a scalar temperature parameter.

Our training objective for fine-tuning is the combination of cross-entropy loss \( \mathcal{L}_{ce} \) and contrastive learning loss \( \mathcal{L}_{cl} \):

\[
\mathcal{L}_{overall} = \lambda \mathcal{L}_{ce} + (1 - \lambda) \mathcal{L}_{cl}, \tag{3}
\]

where \( \lambda \) is a loss scaling hyperparameter.

3.2 Positive Sampling

The strategies of constructing positive samples for contrastive learning have been studied actively. In the following, we give detailed description of two positive sampling strategies for generalizable implicit hate speech detection.

3.2.1 Augmented Post as Positive Samples

It has been shown that unintended biases in a dataset could lead to the generalization issue of a model detecting abusiveness (Wiegand et al., 2019). Due to the lack of lexical cues in implicit hate speech and its subtlety, we suspect that implicit hate speech detector could easily overfit to unintended lexical biases in the dataset. To ease such issue, we suggest using augmented post as a positive sample. Our intuition is that by using augmented variants of posts, which are lexically different but semantically similar with original posts, the model can learn more invariant semantic features.

When we denote augmentation module as \( AUG(\cdot) \), here, we set the positive sample for \( i \)-th input \( x_i \) as \( x_i^{pos} = x_j = AUG(x_i) \). For \( i \leq N \), \( AUG(x_i) \) is the augmented version of \( x_i \). For \( i > N \), \( AUG(x_i) \) is the original input post (be-
We refer to this contrastive learning with augmentation of \( x_i \) using augmented post as a positive sample is defined as:

\[
\mathcal{L}_{cl}^{aug} = -\sum_{i=1}^{2N} \log \frac{e^{h(x_i) - h(AU/G(x_i))/\tau}}{\sum_{k=1}^{2N} 1[\cdot]e^{h(x_i) - h(x_k)/\tau}}.
\]  

We refer to this contrastive learning with augmented posts as \( AugCon \). Then, overall objective for fine-tuning with cross-entropy loss and \( AugCon \) is:

\[
\mathcal{L}_{overall}^{aug} = \lambda \mathcal{L}_{ce} + (1 - \lambda) \mathcal{L}_{cl}^{aug}.
\]  

### 3.2.2 Implication as Positive Samples

Hate speech conveys a targeted group and disparaging stereotypes and biases regarding the group. At times, although presented differently, a group of hateful posts implies similar harmful biases. That is, people generating hate speech often project one implication to various lexical forms of posts. Inspired by the relationship between an implication and its various lexical forms of hateful posts, we propose to use an implication of a hateful post as a positive sample. By pulling a hateful post and its implication in the training process, an implication can work as an anchor for its corresponding hateful posts. This would enable a model to learn the relationship between a hateful post and its concealed meaning, leading to more generalizable implicit hate speech detector.

We assume a module \( IMP(\cdot) \), where we set the positive sample for \( i \)-th input \( x_i \) as \( x_i^{pos} = x_j = IMP(x_i) \). For \( i \leq N \), \( IMP(x_i) \) means an implication of \( x_i \) if \( x_i \) is a hateful post, otherwise (i.e., if \( x_i \) is a non-hateful post) \( IMP(x_i) \) means an augmented version of \( x_i \). For \( i > N \), \( IMP(x_i) \) means the original input post of \( x_i \) (i.e., \( x_i \) is an implication or augmented version of \( IMP(x_i) \)). In detail, for implication, we use implications that are given in IHC and SBIC dataset. For augmentation, we use the same augmentation as \( AugCon \). The contrastive learning loss \( \mathcal{L}_{cl}^{imp} \) using implication as a positive sample is defined as:

\[
\mathcal{L}_{cl}^{imp} = -\sum_{i=1}^{2N} \log \frac{e^{h(x_i) - h(IMC(x_i))/\tau}}{\sum_{k=1}^{2N} 1[\cdot]e^{h(x_i) - h(x_k)/\tau}}.
\]  

We refer to this contrastive learning using implication as \( ImpCon \). Then, overall objective for fine-tuning with cross-entropy loss and \( ImpCon \) is:

\[
\mathcal{L}_{overall}^{imp} = \lambda \mathcal{L}_{ce} + (1 - \lambda) \mathcal{L}_{cl}^{imp}.
\]  

The overview of training a model with \( \mathcal{L}_{overall}^{imp} \) is demonstrated in Figure 2.

### 4 Experiment

#### 4.1 Datasets

We perform binary classification of detecting hateful language on implicit hate datasets. For cross-dataset evaluation, we use three implicit hate speech datasets as \( Hartvigsen et al. (2022) \). Social Bias Inference Corpus (SBIC) (\( Sap et al., 2020 \)) is the dataset with hierarchical annotation of social bias including offensiveness, target, and implied statement. Similarly, Implicit Hate Corpus (IHC) (\( ElSherief et al., 2021 \)) is the implicit hate speech dataset with target and implication collected from hate communities and their followers on Twitter. DynAHate (\( Vidgen et al., 2021 \)) is the hate speech dataset collected through human-and-model-in-the-loop process of deceiving a model.

Since one of our main focus is leveraging implication for generalizable model, we fine-tune models on two datasets with implications, IHC and SBIC. For IHC, we refined the dataset considering the uniformity across annotation stages, resulting in all 'implicit hate' labeled samples having implications. For SBIC, we aggregate annotations of each post. In addition, we merge an implied statement with a target to get an implication following the set of rules in \( Marasović et al. (2022) \).

#### 4.2 Baseline Training Approaches

We experimented with three baseline training approaches.

- **Cross-entropy Loss (CE)**: we fine-tune a model with cross-entropy loss (CE), which is a general approach in hate speech classification.
- **Cross-entropy Loss (CE) with Data Augmentation**: we apply data augmentation to
the training data and train a model using cross-entropy loss. For data augmentation, we use the same augmentation used in AugCon, which substitutes 30% of words with their synonyms using WordNet following Suresh and Ong (2021) \(^2\).

- **Cross-entropy Loss (CE) with Supervised Contrastive Learning**: we fine-tune each model using supervised contrastive learning (SCL) (Gunel et al., 2021) combined with cross-entropy loss. In SCL, posts from the same class are pulled close while others are pushed apart in the representation space \(^3\).

4.3 Implementation Details

We use the pre-trained language model BERT-base-uncased as a base model, since it (and its variants) has shown state-of-the-art performance in hate speech detection (Swamy et al., 2019; Mathew et al., 2021). We also conduct experiments with HateBERT, which shows better generalization ability than BERT in the experiment of Caselli et al. (2021).

We train models for 6 epochs with NVIDIA RTX 3090. For hyperparameter, we search learning rate from \{5e-6, 1e-5, 2e-5, 3e-5, 5e-5\}, temperature \(\tau\) from \{0.1, 0.3, 0.5\}, \(\lambda\) from \{0.25, 0.5, 0.75\} and choose the best model with validation F1 score. We run all experiments on 5 seeds (0, 1, 2, 3, 4) and report the F1 score on the test set.

4.4 Experiment Results

Table 3 and Table 4 shows the cross-dataset evaluation results for the models trained on IHC and SBIC respectively along with in-dataset evaluation results for the models trained on IHC dataset. We use \(\rightarrow\) to distinguish the dataset used for training (on the left) and the dataset used for evaluation (on the right). For example, \(\text{IHC} \rightarrow \text{SBIC}\) means the setting where a model is trained on IHC and then evaluated on SBIC. Boldfaced values on cross-dataset evaluation denote the best performance among different training objectives.

---

### Table 3: Cross-dataset and in-dataset evaluation results for the models trained on IHC dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Objective</th>
<th>IHC → SBIC (Cross-dataset)</th>
<th>IHC → DYNAHATE (Cross-dataset)</th>
<th>IHC → IHC (In-dataset)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>CE</td>
<td>0.568</td>
<td>0.531</td>
<td>0.777</td>
</tr>
<tr>
<td>BERT (Aug)</td>
<td>CE</td>
<td>0.565</td>
<td>0.538</td>
<td>0.777</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + SCL</td>
<td>0.560</td>
<td>0.537</td>
<td>0.777</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + AugCon</td>
<td>0.581</td>
<td>0.546</td>
<td>0.774</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + ImpCon</td>
<td>0.607</td>
<td>0.579</td>
<td>0.780</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + AugCon + ImpCon</td>
<td>0.611</td>
<td>0.577</td>
<td>0.779</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE</td>
<td>0.587</td>
<td>0.547</td>
<td>0.764</td>
</tr>
<tr>
<td>HateBERT (Aug)</td>
<td>CE</td>
<td>0.555</td>
<td>0.528</td>
<td>0.763</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + SCL</td>
<td>0.559</td>
<td>0.528</td>
<td>0.767</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + AugCon</td>
<td>0.584</td>
<td>0.545</td>
<td>0.765</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + ImpCon</td>
<td>0.635</td>
<td>0.594</td>
<td>0.774</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + AugCon + ImpCon</td>
<td>0.630</td>
<td>0.591</td>
<td>0.772</td>
</tr>
</tbody>
</table>

---

### Table 4: Cross-dataset and in-dataset evaluation results for the models trained on SBIC dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Objective</th>
<th>SBIC → IHC (Cross-dataset)</th>
<th>SBIC → DYNAHATE (Cross-dataset)</th>
<th>SBIC → SBIC (In-dataset)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>CE</td>
<td>0.596</td>
<td>0.603</td>
<td>0.838</td>
</tr>
<tr>
<td>BERT (Aug)</td>
<td>CE</td>
<td>0.601</td>
<td>0.604</td>
<td>0.833</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + SCL</td>
<td>0.594</td>
<td>0.610</td>
<td>0.838</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + AugCon</td>
<td>0.597</td>
<td>0.612</td>
<td>0.833</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + ImpCon</td>
<td>0.614</td>
<td>0.612</td>
<td>0.836</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + AugCon + ImpCon</td>
<td>0.596</td>
<td>0.603</td>
<td>0.838</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE</td>
<td>0.587</td>
<td>0.598</td>
<td>0.840</td>
</tr>
<tr>
<td>HateBERT (Aug)</td>
<td>CE</td>
<td>0.591</td>
<td>0.599</td>
<td>0.844</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + SCL</td>
<td>0.593</td>
<td>0.598</td>
<td>0.843</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + AugCon</td>
<td>0.585</td>
<td>0.595</td>
<td>0.841</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + ImpCon</td>
<td>0.599</td>
<td>0.606</td>
<td>0.848</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + AugCon + ImpCon</td>
<td>0.590</td>
<td>0.603</td>
<td>0.843</td>
</tr>
</tbody>
</table>
tion results. We investigate whether AugCon and ImpCon can improve the cross-dataset evaluation performance when combined with cross-entropy loss.

In cross-dataset evaluation, which we mainly focus on, simply adding augmented posts to the training set is not effective. Also, leveraging label information for contrastive learning (SCL) is less effective than our approaches. These results could be attributed to coarse-grained label (only two classes) in our task, which is in line with the results from Suresh and Ong (2021).

Adding AugCon on BERT increases the performance (at most 2.92% improvement) while adding it on HateBERT shows slight decrease. This indicates limited effectiveness of AugCon, particularly when adapted to a domain-shifted pre-trained language model. However, the models trained with ImpCon consistently outperform the models trained only with cross-entropy loss; we obtain at most 9.10% improvement when applied to BERT and 8.71% improvement when applied to HateBERT. This demonstrates the effectiveness of using ImpCon on generalization ability.

We also experimented the combination of AugCon and ImpCon with the same scaling factor between them. Only one result shows 0.58% performance improvement compared to the best performing ImpCon result. We analyze the possible reason in Section 5.1. Regarding the relatively low improvement on the models trained on SBIC, broader definition of class (offensiveness) and thus lower proportion of implication (not all offensive posts have implications) in offensive-labeled posts would be a reason.

For in-dataset evaluation, adding AugCon or ImpCon or combination of them (i.e., AugCon and ImpCon) does not compromise the performance. We note that in-dataset performance can be over-estimated, and cross-dataset evaluation results is rather perceived as better evaluation for measuring generalization ability.

## 5 Analysis

### 5.1 Representation Analysis

We focus on investigating the effect of ImpCon on the representation space. Since ImpCon pulls a paired post-implication in the representation space, we analyze the representation of post-implication pairs quantitatively and qualitatively. Although the model trained with ImpCon would project post-implication pairs of the training set close, it is unknown whether the model can project unseen post-implication pairs close. Hence, we conduct analysis using post-implication pairs in the validation set, which are unseen while training. We use the representation of [CLS] token for the following two analyses. For the uniformity between analyses, we use the same BERT and HateBERT models trained on IHC training set on a seed.

#### Quantitative Analysis

We compute averaged cosine similarity between all post-implication pairs of IHC validation set. As shown in Table 5, two training objectives with ImpCon (CE + ImpCon, CE + AugCon + ImpCon) show higher similarity than others. The similarity gains of ImpCon-based training objectives compared to CE validate that ImpCon enables a model to project unseen post-implication pairs close. While CE + ImpCon shows the highest cosine similarity (0.6752 on BERT, 0.6731 on HateBERT), CE + AugCon + ImpCon shows lower cosine similarity (0.6048 on BERT, 0.5399 on HateBERT). Considering the lowest similarity CE + AugCon showed, AugCon seems to prevent post-implication pairs from being pulled close. We conjecture that this is one of the reasons why simply combining AugCon and ImpCon does not yield the best performance on 3 out of 4 cross-dataset evaluations.

#### Qualitative Analysis

We visualize the learned representation of post-implication pairs from the IHC validation set using t-SNE (van der Maaten and Hinton, 2008). As shown in Figure 3, the representation learned by the training objectives with

<table>
<thead>
<tr>
<th>Model</th>
<th>Objective</th>
<th>Sim.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>CE</td>
<td>0.27</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + AugCon</td>
<td>0.15</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + ImpCon</td>
<td>0.68</td>
</tr>
<tr>
<td>BERT</td>
<td>CE + AugCon + ImpCon</td>
<td>0.60</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE</td>
<td>0.42</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + AugCon</td>
<td>0.17</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + ImpCon</td>
<td>0.67</td>
</tr>
<tr>
<td>HateBERT</td>
<td>CE + AugCon + ImpCon</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Table 5: Quantitative analysis on the representation learned by different training objectives. Using each model fine-tuned with one of the training objectives, we calculated the averaged cosine similarity between all post-implication pairs of IHC validation set.
Figure 3: Visualization of implicit hate posts and implications in IHC validation set using t-SNE. We use BERT model trained on IHC training set with each training objective.

Figure 4: Visualization of three example implications and their corresponding posts using t-SNE. A triangle-marker indicates an implication and a circle-marker indicates a post, respectively. Same colored posts (circle-markers) share the common implication, i.e., triangle-marker colored in the same color. Blue triangle: “Immigrants should be deported”, red triangle: “White people are superior”, and yellow triangle: “Non-white people are inferior”. We use BERT model trained on IHC training set with each training objective.

**ImpCon** (CE + ImpCon, CE + AugCon + ImpCon) shows that the implications and posts are more densely clustered in representation space compared to others, which is in line with our quantitative analysis.

However, it is necessary to investigate whether semantically relevant implications and posts are projected closer than irrelevant ones. As an in-depth analysis, we carefully select some implications and visualize them with their corresponding posts (Figure 4). We choose three implications, where two implications share similar semantic meaning (“White people are superior” and “Non-white people are inferior”) and the other implication has different semantic meaning (“Immigrants should be deported”). We observe that the models with ImpCon-based training objectives (CE + ImpCon, CE + AugCon + ImpCon) produce distinguishable representation between implications with different meanings (blue points are apart from red and yellow points). Also, implications (and their corresponding posts) with similar meaning are densely clustered (red and yellow points are close to each other). The results suggest that semantically relevant implications and posts are projected closer than irrelevant ones in the representation space learned by ImpCon. We believe this is one of the reasons why ImpCon shows improvements on cross-dataset evaluation.

### 5.2 Case Study

We have shown that ImpCon forces relevant implications and posts to be projected closer in the representation space, quantitatively and qualitatively. As a further investigation, we observe whether the model can predict consistently on relevant posts that share the same implication.

We conduct a case study with two groups of hateful posts—one group shares the implication “jews are worthless” and the other group shares “black folks are animals”—in SBIC validation set. We report prediction results using BERT models trained on IHC. In Table 6, the model trained with ImpCon shows consistently correct predictions on posts that share a same implication. For example, for an implication “jews are worthless”, while BERT trained with cross-entropy loss succeeds in predicting correctly on post 1 and 2, it fails to predict correctly on post 3 even though three posts share a same implication. However, the model trained with ImpCon predicts all corresponding posts as offensive correctly.
Table 6: Some implications and their corresponding posts from SBIC validation set. We compare predictions of two BERT models trained on IHC training set with different training objectives (CE and CE + ImpCon). Label 1 denotes there exist offensive contents in the post.

<table>
<thead>
<tr>
<th>Implication 1</th>
<th>Post 1</th>
<th>&quot;How do you turn a Christian into a Jew? Cremate him&quot;</th>
<th>Post 2</th>
<th>&quot;Roses are red, violets are blue, I saw the KKK beating a Jew.&quot;</th>
<th>Post 3</th>
<th>&quot;What is the difference between a jew and a pizza: I don’t like eating pizza&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label: 1</td>
<td>Label: 1</td>
<td>BERT (CE): 1</td>
<td>Label: 1</td>
<td>BERT (CE): 1</td>
<td>Label: 1</td>
<td>BERT (CE): 0</td>
</tr>
<tr>
<td>BERT (CE + ImpCon): 1</td>
<td>1</td>
<td>BERT (CE + ImpCon): 1</td>
<td></td>
<td>BERT (CE + ImpCon): 1</td>
<td></td>
<td>BERT (CE + ImpCon): 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Implication 2</th>
<th>Post 1</th>
<th>&quot;Why are there no black furries? Because they are already animals.&quot;</th>
<th>Post 2</th>
<th>&quot;What’s the difference between KFC and the KKK? Nothing... They both fry animals&quot;</th>
<th>Post 3</th>
<th>&quot;I don’t believe in slavery I support animal rights&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label: 1</td>
<td>Label: 1</td>
<td>BERT (CE): 1</td>
<td>Label: 1</td>
<td>BERT (CE): 0</td>
<td>Label: 1</td>
<td>BERT (CE): 0</td>
</tr>
<tr>
<td>BERT (CE + ImpCon): 1</td>
<td>1</td>
<td>BERT (CE + ImpCon): 1</td>
<td></td>
<td>BERT (CE + ImpCon): 1</td>
<td></td>
<td>BERT (CE + ImpCon): 1</td>
</tr>
</tbody>
</table>

5.3 Error Analysis

We conduct an error analysis on cross-dataset evaluation to facilitate further studies. We provide the confusion matrices (Figure 5) where the models are trained on IHC and evaluated on SBIC. While ImpCon decreased false negatives and false positives, there are still considerable amount of errors. It is notable that 27.39% of predictions are false negatives, which takes a higher proportion than false positives (Figure 5(b)). We inspect such samples, and we suspect a target group that rarely appears in the training set would lead to false negatives. For example, given a hateful post with rare target group anorexic folks, "What do you call an anorexic with a yeast infection? A Quarter-Pounder with Cheese.", the model trained with CE + ImpCon predicts it as a non-offensive post. Since hate speeches on different target groups are based on distinct characteristics (stereotypes) of each group, hate speech on unseen target would limit the generalization ability of the model. Developing a training approach that can generalize well to unseen target groups would be a possible future direction.

6 Conclusions

We study the cross-dataset underperforming problem in implicit hate speech detection task. Empirically, we confirm that the pre-trained language models fine-tuned on an implicit hate speech dataset show relatively low performance on cross-dataset evaluation. We suggest leveraging contrastive learning when fine-tuning implicit hate speech detector to improve generalization ability. Particularly, we propose to utilize shared implication as a positive sample for its corresponding hateful posts, and introduce an implication-based contrastive learning method (ImpCon). Extensive experiments suggest that fine-tuning with ImpCon leads to better generalization ability, resulting in consistent performance improvements on all cross-dataset evaluation with three implicit hate speech datasets.

Acknowledgements

This research was supported by the NRF grant (NRF-2020R1A4A3079947), IITP grant (No. 2021-0-00354) and the AI Graduate School Program (No. 2020-0-01361) funded by the Korea government (MSIT). Han is a corresponding author.
References


Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset


Nils Rethmeier and Isabelle Augenstein. 2022. A primer on contrastive pretraining in language processing: Methods, lessons learned & perspectives. \textit{ACM Computing Surveys (CSUR)}.


A Visualization by t-SNE (HateBERT)

Figure 6: Visualization of implicit hate posts and implications in IHC validation set using t-SNE. We use HateBERT model trained on IHC training set with each training objective.

Figure 7: Visualization of three example implications and their corresponding posts using t-SNE. A triangle-marker indicates an implication and a circle-marker indicates a post, respectively. Same colored posts (circle-markers) share the common implication, i.e., triangle-marker colored in the same color. Blue triangle: “Immigrants should be deported”, red triangle: “White people are superior”, and yellow triangle: “Non-white people are inferior”. We use HateBERT model trained on IHC training set with each training objective.

B Error Analysis (HateBERT)

Figure 8: Confusion matrices for the model predictions on SBIC validation set. We compare the predictions of two HateBERT models trained on IHC training set with (a) CE and (b) CE + ImpCon.