Label-aware Hard Negative Sampling Strategies with Momentum Contrastive Learning for Implicit Hate Speech Detection

Warning: This paper contains examples that can be offensive or upsetting.

Jaehoon Kim[1], Seungwan Jin[2], Sohyun Park[1], Someen Park[1], Kyungsik Han[1,2,*]
[1] Department of Artificial Intelligence, Hanyang University, Seoul, Republic of Korea
[2] Department of Data Science, Hanyang University, Seoul, Republic of Korea
{jaehoonkimm,seungwanjin,sohyunpark,someeeen,kyungsikhan}@hanyang.ac.kr

Abstract

Detecting implicit hate speech that is not directly hateful remains a challenge. Recent research has attempted to detect implicit hate speech by applying contrastive learning to pre-trained language models such as BERT and RoBERTa, but the proposed models still do not have a significant advantage over cross-entropy loss-based learning. We found that contrastive learning based on randomly sampled batch data does not encourage the model to learn hard negative samples. In this work, we propose **La**bel-aware **H**ard **N**egative sampling strategies (**LAHN**) that encourage the model to learn detailed features from hard negative samples, instead of naive negative samples in random batch, using momentum-integrated contrastive learning. LAHN outperforms the existing models for implicit hate speech detection both in- and cross-datasets. The code is available at https://github.com/Hanyang-HCC-Lab/LAHN

1 Introduction

Online hate speech has become one of the major social problems as it leads to discrimination against certain groups and social conflict (Howard, 2019; Matamoros-Fernández and Farkas, 2021). Hate speech can be either explicit, which directly uses hateful language, or implicit, which metaphorically implies hateful language (Waseem et al., 2017). While explicit hate speech can be addressed relatively easily through the use of automated filters (Watanabe et al., 2018; Xiang et al., 2012; Rodríguez-Sánchez et al., 2020), dealing with implicit hate speech is highly challenging. For example, identity term bias (ElSherief et al., 2021; Dixon et al., 2018), where identity terms (e.g., black, jew) frequently appear in hateful contexts, can over-bias the model and cause false positives in implicit hate speech detection. Given the characteristics of implicit hate speech, it will be important to encourage

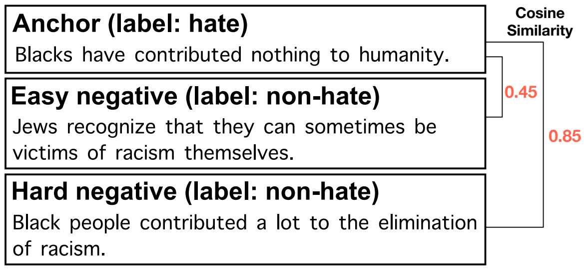

Figure 1: Our research motivation. Easy negatives have a low semantic similarity to anchors, and hard negatives have a high semantic similarity to anchors.

models to learn subtle differences between similar sentences that might otherwise confuse them. Research has built implicit hate speech datasets of implicit hate speech (ElSherief et al., 2021; Sap et al., 2020; Hartvigsen et al., 2022; Vidgen et al., 2021) and proposed detection models (Kim et al., 2022, 2023) using contrastive learning; however, the model showed limited performance improvement in in-dataset evaluation or have limitations that require external knowledge or additional computational resources for pre-training.

Figure 1 shows an easy negative (middle) and a hard negative (bottom) for an anchor sentence targeting black. Easy negatives have opposite labels and are relatively distinct from the anchor, meaning that for the model it is not difficult to distinguish from the anchor. On the other hand, hard negatives are semantically similar to the anchor but have opposite labels, meaning that the model may have difficulty distinguishing them from the anchor. The key task is to effectively identify hard negatives and train the model accordingly. If the negative sample trained along with the anchor is similar to the anchor, the model can be trained to better distinguish relatively indistinguishable data (Robinson et al., 2021; Jiang et al., 2022; Wu et al., 2021). However, existing work has used naive contrastive learning, which encourages contrast between data in randomly sampled mini-batches to learn repre-

*Corresponding author

Findings of the Association for Computational Linguistics: ACL 2024, pages 16177–16188
August 11-16, 2024 ©2024 Association for Computational Linguistics

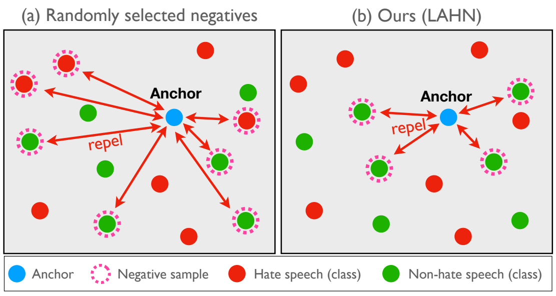

Figure 2: Illustration of the two methods using contrastive learning in a situation where the class of the anchor is hate speech. (a) The random sampling method randomly selects negative samples. (b) LAHN selects only different classes from the anchor as negative samples (i.e., the green dots include only the opposite class from the anchor).

sentations and fails to guarantee the learning of hard negative samples.

In this work, we propose a novel approach to implicit hate speech detection, namely **Lab**el-aware **H**ard **N**egative sampling strategies (**LAHN**). LAHN focuses more on distinguishing between the anchor and hard negatives, mitigating overfitting to the context of the text or specific words. Inspired by MoCo (He et al., 2020), LAHN employs a momentum queue to effectively expand the negative samples that are candidates for hard negative samples, and compactly performs contrastive learning by extracting the top-k hard negatives that require sophisticated disentangling from the anchor. A key difference from previous research is that LAHN extracts hard negatives from the momentum queue in contrastive learning based on the similarity between the anchor and the negatives and the level of ambiguity for each negative (Figure 2).

In summary, our contributions are as follows:

- We propose LAHN, a novel method that focuses on hard negatives that should be disentangled from the anchor to promote the effective learning of hate speech representations with implicit characteristics.
- Contrary to previous studies, we observe that LAHN can improve performance in both in- and cross-dataset evaluation with a simple dropout noise augmentation without external knowledge and additional cost.
- We validate the generalized learning effect of LAHN by achieving state-of-the-art performance in both in- and cross-dataset evaluation on four representative public benchmark datasets for implicit hate speech detection.

2 Related work

2.1 Implicit Hate Speech Detection

Implicit hate speech refers to expressions of hatred or discrimination that are not directly stated or overtly aggressive but are conveyed through subtle, indirect language, insinuations, or coded messages (ElSherief et al., 2021; Kim et al., 2023). According to prior studies (Ocampo et al., 2023; Yang et al., 2023), it is challenging to identify implicit hate speech due to its reliance on background knowledge, cultural context, and the ability to infer the implied meanings behind seemingly neutral or ambiguous expressions. For instance, Ocampo et al. (2023) explores the absence of consistent hate-speech-specific prosody across languages, indicating the importance of linguistic and cultural nuances in understanding hate speech.

For this reason, the existing pre-trained language models in the hate speech domain, such as HateBERT (Caselli et al., 2021) or fBERT (Sarkar et al., 2021), may have a spurious correlation issue that classifies an input text as hateful due to the presence of specific identity terms (e.g., Black, Asian, etc.) and this hinders generalized performance. This body of work underscores the need for sophisticated, context-aware approaches in hate speech detection systems to effectively address the subtleties of implicit hate speech, considering its varied manifestations and significant societal impact.

Contrastive learning methods (Wu et al., 2020; He et al., 2020; Khosla et al., 2020; Giorgi et al., 2021; Gao et al., 2021) has been increasingly recognized as a necessary and effective approach for detecting implicit hate speech due to its ability to handle the context-dependent nature of such content by leveraging the subtle differences and similarities between hateful and non-hateful content, enhancing model sensitivity to the nuanced expressions of hate speech. Kim et al. (2022) proposed ImpCon, which uses the external knowledge (i.e., the implication of anchor sentences) as positive samples using contrastive loss for implicit hate speech detection. The recently proposed ConPrompt (Kim et al., 2023) utilized machine-generated statements to improve implicit hate speech performance by applying contrastive learning using example sentences from the original prompt as positive samples. However, these previous methods still have limitations due to reliance on pre-defined external knowledge or text generation costs.

Recently, there has been research in using Large

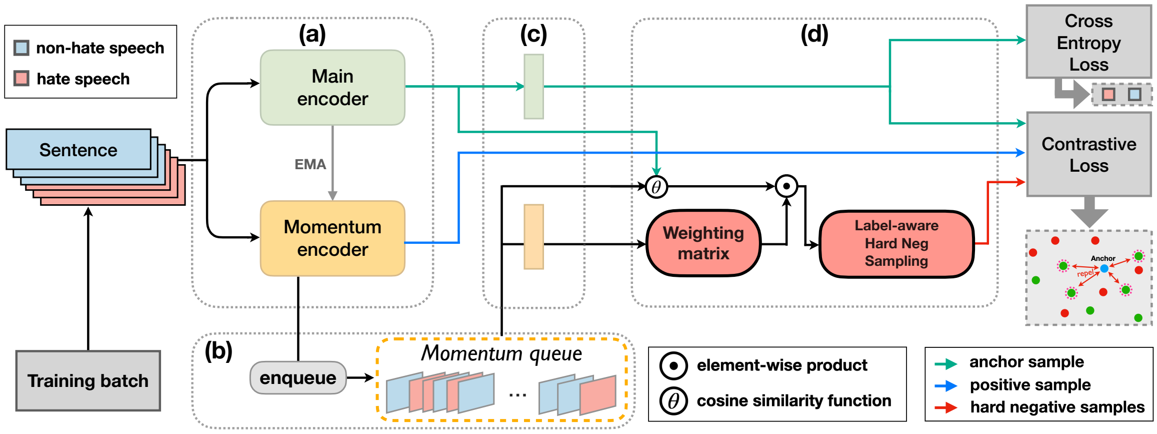

Figure 3: The overview of our LAHN. (a) shows the Momentum encoder being updated via EMA based on the main encoder. (b) enqueues the features extracted by the Momentum encoder. (c) is the prediction head of each encoder, which returns the prediction logits of the input features. (d) weights the true negatives in the momentum queue based on the prediction probability obtained from (c) and samples the hard negatives among them. ($\odot$: element-wise product function, θ: cosine similarity function.)

Language Models (LLMs) for hate speech detection. Zhang et al. (2024) investigated the sensitivity and limitations of LLMs for performing hate speech detection, and Roy et al. (2023) explored several prompting strategies to improve the detection capabilities of LLMs. Yang et al. (2023) proposed the HARE framework, which incorporates explanations generated using chain-of-thoughts (CoT) into the language model learning process. Despite these efforts, the performance of LLMs in hate speech detection is still limited, and costly compared to the language models such as BERT and RoBERTa. Our experimental results on implicit hate speech detection using LLMs are presented in Appendix A.

2.2 Hard Negative Mining

Hard negative mining is an important technique in various areas of machine learning that significantly improves a model's performance by carefully selecting negative samples that are less obviously distinguishable from positive samples (Robinson et al., 2021; Gunel et al., 2020; Schroff et al., 2015; Wu et al., 2021; Ge et al., 2021; Kalantidis et al., 2020). In contrastive learning, many studies have found that focusing on hard negative samples among negative samples learns better representations and improves performance (Robinson et al., 2021; Gunel et al., 2020; Schroff et al., 2015; Kalantidis et al., 2020).

In the field of computer vision, many studies related to hard negatives have been proposed. Ge et al. (2021) identified the impact of hard negatives in learning strategies to generate appropriate negative samples, and Robinson et al. (2021) proposed H-SCL, which uses the label information of samples to introduce hard negative sampling in supervised contrastive learning. Wu et al. (2021) proposed a ring method to step-wise retrieve hard negative samples that are neither too hard nor too easy, and experiments showed that dynamically changing the hard negative sampling range can help to learn.

In NLP, hard negative mining can help models learn finer distinctions between textual data and better understand the nuances of language, context, and semantic meaning. The supervised SimCSE for learning text representations proposed by Gao et al. (2021) considered predefined contradiction pairs as hard negatives for the first time in the field of NLP and encouraged learning to distinguish hard negatives in contrastive learning. However, supervised SimCSE relies on predefined in-batch hard negatives and only utilizes samples within the batch in contrastive learning, so it is limited in considering negative samples broadly. ESimCSE (Wu et al., 2022) and MoCoSE (Cao et al., 2022), which were proposed to compensate for these limitations of SimCSE, integrated momentum contrastive learning with SimCSE, but hard negative samples were not considered in the contrastive learning process.

To the best of our knowledge, no research has explored how to effectively integrate hard negatives with supervised contrastive learning using label information without relying on external knowledge. In this paper, we propose LAHN, which integrates

Algorithm 1: Pseudocode of LAHN

```
# E, E_m: main encoder end momentum encoder
# queue: momentum contrast queue
# m, t: momentum and temperature parameter
# k: hard negative sampling size
# sim: cosine similarity function

# load a batch x with N samples
for x in loader:
  x_anc, pred = E.forward(x)
  x_aug, _ = E_m.forward(x)
  x_aug = x_aug.detach() # no gradient

  # enqueue the current batch embeddings
  enqueue(queue, x_aug)
  # dequeue the earliest batch embeddings
  dequeue(queue)

  # hard negative sampling for anchor
  _, weights = E_m.forward(queue)
  negs = sort(sim(x_anc, queue) * weights)
  hard_neg = topk(negs, k) # top-k sampling
  # extract pos and neg simialrity
  pos = sim(x_anc, x_aug)/t
  pos = diag(pos).view(-1, 1) # extract pos
  neg = sim(x_anc, hard_neg)/t
  logits = concat([pos, neg], dim=1)

  # Contrastive loss and CE loss
  labels = Zeros(logits)[:, 0] = 1
  loss_1 = CrossEntropy(logits, labels)
  loss_2 = CrossEntropy(pred, x.labels)
  (loss_1 + loss_2).backward()
  update(E.params)
  # momentum update
  E_m.params = m*E_m.params+(1-m)*E.params
```

momentum contrastive learning and label-aware hard negative sampling strategies to effectively handle implicit hate speech data.

3 Method

As shown in Figure 2-(a), the existing method performs contrastive learning on all other samples with equal weight, including those of the same class, for a given anchor in a mini-batch. This characteristic makes it difficult for the model to focus on distinguishing between the anchor and hard negatives, which are semantically similar and embedded close to the anchor.

On the other hand, our LAHN first uses label information to identify negatives with labels different from the anchor's, which should be embedded in opposite directions. Next, the momentum encoder calculates the probability that the negative is in the same class as the anchor, and this probability is used to multiply the similarity value between the negative and the anchor. Finally, LAHN extracts the top-k hard negatives based on the calculated values and performs compact contrastive learning for the given anchor (Figure 2-(b)). Through this, LAHN better captures the detailed differences between implicit hate speech and non-hate speech by focusing on hard negatives.

Figure 3 illustrates the overview of our LAHN. Random mini-batch samples are used as the input to the main and momentum encoders, and the output embedding of the momentum encoder is inserted into the momentum queue. Hard negative sampling for each anchor is performed after the momentum queue is filled with at least a quarter of the size. As in MoCo (He et al., 2020), we use an exponential moving average (EMA) to slowly update the momentum encoder based on the weights of the main encoder. Algorithm 1 presents the pseudocode implementation.

3.1 Label-aware Hard Negative Sampling

In this section, we propose a label-aware hard negative sampling strategy to effectively integrate label information and hard negative mining methods in supervised contrastive learning. We assume that focusing on contrastive learning to distinguish the anchor from the hard negatives that are semantically similar to the anchor can help learn disentangled representations of implicit hate speech data.

In supervised contrastive learning, false negatives, where samples with the same label as the anchor are used as negative samples, can hinder the representation learning of the model (Kalantidis et al., 2020). To exclude false negative cases in the hard negative sampling process, we identified the true negatives of the momentum queue using the label information of each anchor in the batch and then sorted them by finding the cosine similarity between anchors and true negatives.

To sample hard negatives, which should be primarily included for successful compact contrastive learning, we considered not only the contextual similarity to the anchor, but also the ambiguity of the representation to classify the class of the hard negative. First, we calculated the probability that all true negatives in the momentum queue, which are hard negative candidates for contrastive learning, are predicted as the class of the anchor through the prediction head of the momentum encoder. Second, we applied hard negative sampling based on the assumption that the closer the probability of being predicted as the anchor class for true negatives is to 1 (more ambiguity), the more sophisticated representation learning is required. Finally, we multiplied the prediction probabilities by the similarity values with the anchor, sampled the top-k hard negatives based on the multiplied values, and assigned them as negatives of each anchor in compact contrastive learning.

3.2 Training objective for Implicit Hate Speech Detection

In this section, we conduct contrastive learning for disentangled representation learning and classification training for implicit hate speech detection. First, we employ momentum contrastive learning (He et al., 2020) to maximize the effectiveness of our proposed hard negative sampling strategy. In the implicit hate speech domain, text augmentations, such as the most commonly used token replacement (e.g., synonym substitution), can alter a sentence's hateful or non-hateful nature. Therefore, we utilized the augmentation technique that employs dropout noise, as used in SimCSE (Gao et al., 2021). In Section 5, the experiments using external knowledge employed the augmentation techniques (i.e., implication on hate speech and synonym substitution on non-hate speech) used in (Kim et al., 2022). After hard negative sampling in Section 3.1, we used InfoNCE (Oord et al., 2018) loss for the contrastive learning as follows:

$$\mathcal{L}_{\text{CL}} = -\log \frac{\exp(\text{sim}(x_i, x_i^p))/\tau}{\sum_{j=1}^{N} \exp(\text{sim}(x_i, x_j^n))/\tau} \quad (1)$$

where x_i is the anchor sample of the batch, and every anchor has one positive sample x^p. The proposed method uses only the batch's positive samples and the hard negatives sampled from the momentum queue as negatives. N denotes the number of selected hard negatives from the queue, x^n denotes the negative samples, τ is a temperature parameter, and sim is the cosine similarity function. To minimize the loss in the corresponding contrastive learning process, the similarity between anchor and positive samples should be maximized, and the similarity between anchor and negative samples should be minimized. The loss function for implicit hate speech detection is as follows.

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)] \quad (2)$$

where $\hat{y}_i \in \{0, 1\}$ is one-hot encoded ground truth. The final loss that we use for training is as follows:

$$L = (1 - \lambda)L_{CL} + \lambda L_{CE} \quad (3)$$

where λ is a loss scaling parameter and we empirically found and set the parameter λ as 0.1.

4 Experiment

4.1 Datasets

Similar to the approach in (Kim et al., 2022; Hartvigsen et al., 2022), we used the implicit hate speech benchmark datasets—IHC, SBIC, DynaHate, and ToxiGen—for model evaluation.

- **Implicit Hate Speech Corpus** (ElSherief et al., 2021): The dataset focuses on implicit hate speech collected from Twitter. It comprises 19,112 tweets, with 4,909 labeled as implicit hate and 933 labeled as explicit hate.
- **Social Bias Inference Corpus** (Sap et al., 2020): The dataset with hierarchical categories of social biases and stereotypes. It supports large-scale modeling and evaluation with 150k structured annotations of social media posts, covering over 34k implications about demographic groups.
- **Dynamically Generated Hate Speech Dataset** (Vidgen et al., 2021): The dataset comprises 41,255 entries generated by a human-and-model-in-the-loop process. It captures hate speech against the ten most frequently targeted groups, including black people, women, and Jews, among others.
- **ToxiGen** (Hartvigsen et al., 2022): Created using GPT-3, this dataset contains 274,186 machine-generated statements, with over 135k toxic and 135k benign statements. It targets 13 minorities, such as Blacks, Jews, and LGBTQ+ people.

4.2 Implementation Details

We used a pre-trained language model (RoBERTa-base (Liu et al., 2019) and BERT-base-uncased (Devlin et al., 2019)) as a sentence encoder in all experiments and used the Adam optimizer with a learning rate of 2e-5, batch size of 16, and dropout of 0.1 during the fine-tuning process. We used NVIDIA RTX 4090 GPU (24GB) for training all models, and the hyper-parameters were set to loss scaling parameter $\lambda \in \{0.1\}$, momentum parameter $m \in \{0.999\}$ based on previous studies, and the temperature parameter $\tau \in \{0.05, 0.07, 0.1\}$, momentum queue size $q \in \{512, 1024, 2048\}$, and hard negative $k \in \{16, 32, 64\}$ for further hyper-parameter search. We chose the best model score with macro F1-score in the validation.

Table 1: To investigate both case with/without external knowledge, we employ two datasets (IHC, SBIC) with implication information as the training dataset among the public benchmark dataset. The top table shows the results of the in-dataset (IHC) evaluation and the cross-dataset (SBIC, DynaHate, ToxiGen) evaluation. The bottom table contains the results of the in-dataset (SBIC) evaluation and the cross-dataset (IHC, DynaHate, ToxiGen) evaluation.

Training Dataset	Pre-trained Language Model	External Knowledge	Objective	Evaluation Dataset				Average
				IHC	SBIC	DynaHate	ToxiGen	
IHC	BERT-base	✗	Cross-Entropy Loss (CE)	77.49	57.05	53.69	60.80	62.26
		✗	SCL (Gunel et al., 2020)	77.81	59.19	55.84	62.19	63.76
		✗	**LAHN (ours)**	**78.40**	**62.83**	**57.80**	**63.21**	**65.56**
		✓	ImpCon (Kim et al., 2022)	78.39	54.55	**59.41**	59.64	63.00
		✓	**LAHN (ours)**	**78.62**	**62.02**	56.13	**62.92**	**64.92**
	RoBERTa-base	✗	Cross-Entropy Loss (CE)	79.95	55.03	47.34	59.35	60.42
		✗	SCL (Gunel et al., 2020)	79.33	57.77	46.92	60.89	61.23
		✗	**LAHN (ours)**	**80.11**	**60.57**	**48.46**	**63.94**	**63.27**
		✓	ImpCon (Kim et al., 2022)	78.78	63.82	**50.13**	61.79	63.63
		✓	**LAHN (ours)**	**80.58**	**64.01**	49.54	**64.49**	**64.66**

Training Dataset	Pre-trained Language Model	External Knowledge	Objective	Evaluation Dataset				Average
				IHC	SBIC	DynaHate	ToxiGen	
SBIC	BERT-base	✗	Cross-Entropy Loss (CE)	59.47	83.72	60.17	67.54	67.73
		✗	SCL (Gunel et al., 2020)	60.07	**84.14**	60.97	67.62	68.20
		✗	**LAHN (ours)**	**62.36**	83.98	**63.06**	**69.58**	**69.75**
		✓	ImpCon (Kim et al., 2022)	58.64	83.53	59.50	66.54	67.05
		✓	**LAHN (ours)**	**61.58**	**84.31**	**60.97**	**68.52**	**68.85**
	RoBERTa-base	✗	Cross-Entropy Loss (CE)	59.68	85.27	61.62	68.54	68.78
		✗	SCL (Gunel et al., 2020)	59.61	85.25	61.17	68.77	68.70
		✗	**LAHN (ours)**	**64.74**	**85.45**	**64.32**	**70.61**	**71.28**
		✓	ImpCon (Kim et al., 2022)	56.95	84.66	60.70	66.77	67.27
		✓	**LAHN (ours)**	**64.05**	**85.80**	**63.26**	**69.91**	**70.76**

4.3 Baseline Models

- **Cross-Entropy (CE) loss**: CE loss is widely adopted as a general approach to classification tasks and hate speech detection.
- **Supervised Contrastive Learning (SCL) with CE loss** (Gunel et al., 2020): This method enhances CE loss by combining supervised contrastive learning. This is effective for complex tasks such as hate speech detection, where distinguishing subtle class differences.
- **Contrastive Learning using Implication (ImpCon) with CE loss** (Kim et al., 2022): This method improves CE loss by integrating implication-based contrastive learning. ImpCon improves the model to understand contextual relationships by injecting common implications of implicit hate speech.

5 Results and Analysis

5.1 Implicit Hate Speech Detection Results

Table 1 shows the evaluation results in the four datasets for the training model on the IHC and SBIC datasets. External knowledge means that additional training knowledge beyond the existing data is used, such as implication and synonym substitution. In the case without external knowledge, the encoder dropout noise (Gao et al., 2021) is used as the positive sample. In the cases with external knowledge (Ours, ImpCon), implication was used as the positive sample for hate speech, and synonym substitution was used as the positive sample for non-hate speech, the same as Kim et al. (2022).

In-dataset evaluation (i.e., IHC training set $\rightarrow$ IHC test set, SBIC training set $\rightarrow$ SBIC test set) and cross-dataset evaluation (i.e., IHC training sett $\rightarrow$ all test sets except IHC test set, SBIC training sett $\rightarrow$ all test sets except SBIC test set), LAHN achieves the best performance in 29 out of 32 comparison cases. Furthermore, unlike previous studies (Kim et al., 2022) that found inferior or equal performance to single CE loss learning in the in-dataset case, LAHN achieves better performance than CE loss learning in all in-dataset evaluations.

LAHN is also robust in the absence of external knowledge. We observe that SCL loss without external knowledge performs poorly relative to CE loss, and Kim et al. (2022) shows the same trend for SCL with augmentation based on synonym substitution. In contrast, LAHN, without external knowledge, which uses only dropout noise applied to anchors as a positive sample, achieves the best performance in 15 out of 16 cases without external knowledge comparison.

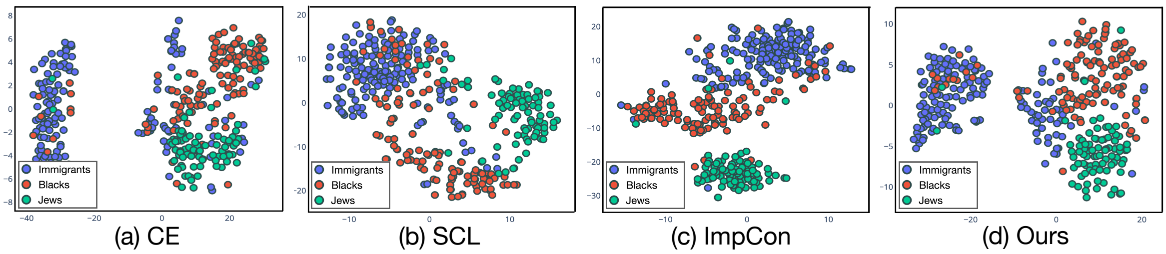

Figure 4: Visualization of implicit hate speech on three targets with the IHC dataset in the In-dataset setting. (Blue: hate speech targeted Immigrants, Red: hate speech targeted Blacks, Green: hate speech targeted Jews.)

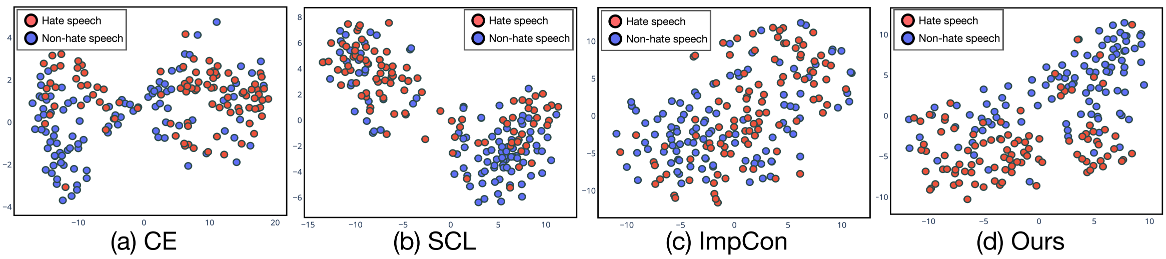

Figure 5: Visualization of implicit hate speech and non-hate sentences that targeted Blacks with the ToxiGen dataset in the zero-shot setting. (Blue: Non-hate speech, Red: Implicit hate speech.)

In the case using the SBIC training dataset, the average score of LAHN without external knowledge (Average column) outperforms the average score of LAHN with external knowledge in both BERT and RoBERTa models. This indicates that the implication used as external knowledge is of poor quality or that the implication may lead to negative bias in model training. The average scores for the cases without external knowledge using BERT and RoBERTa are 69.75 and 71.28, respectively, while the cases with external knowledge are lower (68.85 and 70.76). This shows that the implication information used as external knowledge for SBIC's implicit hate speech data or the synonym substitution augmentation used as external knowledge for non-hate speech is likely to be of poor quality.

5.2 Qualitative Analysis

Figure 4 shows a visualization of some implicit hate speech data from the IHC validation set using t-SNE (Van der Maaten and Hinton, 2008) on a model trained on the IHC dataset by three targets (Immigrants, Blacks, Jews).

SCL (Figure 4-b) shows more dense clusters compared to CE (a) but still has some samples mixed in with each other. ImpCon and LAHN, on the other hand, form clearer cluster boundaries. We note that LAHN forms different sub-clusters within the immigrant cluster. SCL and ImpCon were trained to cluster similar classes using label information and implication, respectively. They push or pull semantically similar or dissimilar samples within randomly sampled mini-batches, depending on the training strategy. In this process, contrastive loss can degrade the semantic similarity of a representation by further attracting positive samples that are already close enough or pushing negative samples that are already far enough away.

In contrast, our method assumes that the pre-trained language model already has a high-quality representation and encourages the model to use only hard negatives for training in order to increase the classification performance of the model without compromising the semantic information as much as possible. As shown in Figure 4-d, LAHN ensures the margin between sentences with different target information while maintaining the margin of data with different semantic characteristics within the immigrants.

Figure 5 shows a visualization of the extracted embeddings for the part of the ToxiGen dataset in the zero-shot setting with the model fine-tuned on the IHC dataset. We randomly sampled both hate speech and non-hate speech about black people. The embeddings using LAHN form a relatively sharper boundary between hate speech and non-hate speech in compared to the other three models. This result shows that our LAHN leads the model to form a more generalizable representation compared to the other methods.

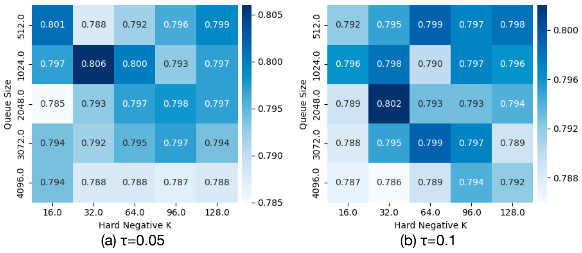

Figure 6: The impact of hyper-parameters q (queue size) and k (sampling size) on IHC dataset with RoBERTa-base model.

5.3 Effect of Hyperparameters

We evaluate the effects of the hyper-parameters q (queue size) and k (hard negative sample size). Cao et al. (2022) found that for NLP tasks where the model has a fast update rate, a queue size as high as 4096 can cause performance degradation. This shows that, unlike MoCo, queue expansion does not inevitably lead to better performance. This finding is consistent with our results, and Figure 6 shows the variation of the model's performance with queue size and hard negative sample size. The x-axis is the number of hard negatives sampled for each anchor, and the y-axis is the size of features that can be candidates for hard negatives.

In contrast to previous work (Cao et al., 2022; Wu et al., 2022), we did not use all of the queue features as negative samples but only sampled some of them, meaning that an increase in queue size does not imply the use of outdated information. Therefore, an increase in queue size does not necessarily lead to a linear decrease in performance. Nevertheless, our ablation results show that performance degradation occurs at a certain point in the queue size. That is, as the queue size increases, outdated samples may also become candidates for hard negative sampling, which can negatively affect performance. This result shows that maintaining a large number of hard negative sample candidates does not always improve performance.

Similarly, we found that increasing the number of hard negative samples does not result in a linear performance improvement. Since our method determines the hard negative samples based on the sorted similarity between the anchor and the negative sample, selecting a large number of hard negative samples implies that a high similarity to the anchor cannot be guaranteed. Therefore, our results show that it is necessary to experimentally select the appropriate number of hard negative samples and queue size.

Table 2: Ablation study results for in-dataset evaluation of our methods (w/ RoBERTa-base) on IHC and SBIC.

Components			IHC		SBIC		Average
MoCo	**HN-Samp**	**S-Weight**	**Acc**	**F1**	**Acc**	**F1**	
✗	✗	✗	83.12	79.08	85.72	85.39	83.33
✓	✗	✗	83.47	78.99	84.93	84.69	83.02
✓	✓	✗	83.58	79.95	85.40	84.99	83.48
✓	✓	✓	**84.36**	**80.58**	**86.06**	**85.80**	**84.20**

5.4 Ablation Study

We conducted the ablation studies by removing each component of our methods on the IHC dataset and the SBIC dataset with RoBERTa-based model.

- **MoCo**: Momentum Contrast means that all samples in the Momentum Contrast queue with label information are used during contrastive learning.
- **HN-Samp**: Hard Negative Sampling means to use the Label-aware Hard Negative Sampling strategy with the momentum queue.
- **S-Weight**: Similarity Weight means using the prediction probability from the momentum encoder to induce the model to select features with a high probability of confusion as hard negatives.

The results are shown in Table 2. Because of the interdependence of our methods, a total of four progressive ablation studies were performed (i.e., HN-Samp cannot be used alone without SupMoCo, and S-Weight cannot be used alone without HN-Samp). These results show that LAHN can significantly increase all performance compared to the MoCo, including false negatives to increase the number of negative samples, or HN-Samp which only similarity-based hard negative sampling without S-weight.

6 Conclusion and Future Work

In this paper, we propose LAHN that incorporates the momentum queue to extract hard negatives, which are more likely to be confused by the model as compared to anchors. We demonstrated the effectiveness of LAHN in both in-dataset and cross-dataset performance evaluations, compared to existing methods. In the future, we will focus on finding new sampling metrics that are more advanced than similarity-based hard negative sampling for implicit hate speech detection and finding ways to exploit better semantic features from hard negatives.

7 Limitations

While our proposed LAHN has demonstrated its effectiveness in implicit hate speech detection, our research has some limitations.

First, LAHN relies on supervision, which reduces the advantage of MoCo in allowing a large amount of unlabeled data to be used for supervised learning. In addition, the dual encoder framework, which uses an additional MoCo encoder, still has cost limitations compared to other deep learning methods.

Second, our strategy is limited in that it requires the exploration of a large number of hyperparameters. As we have seen in Section 5.4, LAHN exhibits dynamic performance depending on hyperparameters such as q, k, and τ, which force the cost of hyperparameter exploration to vary across the dataset.

Third, since we sampled only a fraction of the MoCo queue, we may lose the benefit of MoCo's use of large, consistent negative samples for training. In addition, since we are not using a large queue size like MoCo, it may be more advantageous to adopt an in-batch hard negative sampling strategy. However, we emphasize that our methodology overcomes the constraints of limited computational resources, which may inspire future work.

In future work, we can explore methods that are robust to multiple hyper-parameters, and investigate strategies for effectively sampling out-of-batch hard negatives for training without introducing MoCo.

8 Ethics Statement

Our work aims to extend previous research on implicit hate speech detection and contribute to solving social conflicts caused by hateful content. We use publicly available open datasets containing implicit hate speech without compromising user privacy.

Implicit hate speech detection models can still be biased, and our method may cause the risk of increasing false predictions that have not been observed in previous studies and are not revealed by the performance. However, we believe that our ongoing efforts to improve implicit hate speech detection can help mitigate this risk.

Acknowledgements

This research was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2020-II201373, Artificial Intelligence Graduate School Program(Hanyang University)), and the National Research Foundation of Korea (NRF) (2018R1A5A7059549, 2021S1A5A2A03065899).

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Rui Cao, Yihao Wang, Yuxin Liang, Ling Gao, Jie Zheng, Jie Ren, and Zheng Wang. 2022. Exploring the impact of negative samples of contrastive learning: A case study of sentence embedding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3138–3152.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. Hatebert: Retraining bert for abusive language detection in english. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. 2021. Robust contrastive learning using negative samples with diminished semantics. *Advances in Neural Information Processing Systems*, 34:27356–27368.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. Declutr: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Jeffrey W Howard. 2019. Free speech and hate speech. *Annual Review of Political Science*, 22:93–109.

Ruijie Jiang, Thuan Nguyen, Prakash Ishwar, and Shuchin Aeron. 2022. Supervised contrastive learning with hard negative samples. *arXiv preprint arXiv:2209.00078*.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Youngwook Kim, Shinwoo Park, and Yo-Sub Han. 2022. Generalizable implicit hate speech detection using contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6667–6679.

Youngwook Kim, Shinwoo Park, Youngsoo Namgoong, and Yo-Sub Han. 2023. Conprompt: Pre-training a language model with machine-generated data for implicit hate speech detection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10964–10980.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ariadna Matamoros-Fernández and Johan Farkas. 2021. Racism, hate speech, and social media: A systematic review and critique. *Television & New Media*, 22(2):205–224.

Nicolas Benjamin Ocampo, Ekaterina Sviridova, Elena Cabrio, and Serena Villata. 2023. An in-depth analysis of implicit and subtle hate speech messages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1997–2013. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *International Conference on Learning Representations (ICLR)*.

Francisco Rodríguez-Sánchez, Jorge Carrillo-de Albornoz, and Laura Plaza. 2020. Automatic classification of sexism in social networks: An empirical study on twitter data. *IEEE Access*, 8:219563–219576.

Sarthak Roy, Ashish Harshvardhan, Animesh Mukherjee, and Punyajoy Saha. 2023. Probing llms for hate speech detection: strengths and vulnerabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6116–6128.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490.

Diptanu Sarkar, Marcos Zampieri, Tharindu Ranasinghe, and Alexander Ororbia. 2021. fbert: A neural transformer for identifying offensive content. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1792–1798.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84.

Hajime Watanabe, Mondher Bouazizi, and Tomoaki Ohtsuki. 2018. Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE access*, 6:13825–13835.

Mike Wu, Milan Mosse, Chengxu Zhuang, Daniel Yamins, and Noah Goodman. 2021. Conditional negative sampling for contrastive learning of visual representations. In *International Conference on Learning Representations*.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022. Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. 2012. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1980–1984.

Yongjin Yang, Joonkee Kim, Yujin Kim, Namgyu Ho, James Thorne, and Se-Young Yun. 2023. Hare: Explainable hate speech detection with step-by-step reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5490–5505.

Min Zhang, Jianfeng He, Taoran Ji, and Chang-Tien Lu. 2024. Don't go to extremes: Revealing the excessive sensitivity and calibration limitations of llms in implicit hate speech detection. *arXiv preprint arXiv:2402.11406*.

A Implicit Hate Speech Detection on Large Language Models

A.1 Implementation Details

To evaluate the performance of Large Language Models on implicit hate speech detection, we employ four LLMs (GPT-3.5-turbo-0125 from OpenAI, Claude-3-Haiku-20240307 from Anthropic, and Llama3-8B-Instruct-v1 and Llama3-70B-Instruct-v1 from Meta). We used the OpenAI API [1] for the GPT-3.5-turbo-0125, and used Bedrock on Amazon Web Services [2] for the other models.

[1] https://openai.com/api/

[2] https://aws.amazon.com/bedrock/

Table 3: The performance of four Large Language Models using Zero-Shot (ZS) and Few-Shot (FS) prompts on two evaluation datasets (IHC, SBIC).

Model	Method	Evaluation Dataset	
		IHC	SBIC
GPT-3.5-turbo-0125	ZS	72.02	73.99
	FS	75.30	76.10
Claude-3-Haiku-20240307	ZS	15.83	26.71
	FS	72.20	67.77
Llama3-8B-Instruct-v1	ZS	37.35	37.62
	FS	73.22	72.36
Llama3-70B-Instruct-v1	ZS	73.86	57.82
	FS	76.98	73.15

A.2 Experimental Results

Table 3 shows the performance of the LLMs for the implicit hate speech detection on two datasets (IHC, SBIC). Compared to traditional language models, LLMs with much larger parameters require enormous resources for fine-tuning. Therefore, in-context learning is essential for the efficient use of LLMs (Dong et al., 2022). Zero-shot learning, which provides only a description of the task, and Few-shot learning, which provides examples of the task, are the typical in-context learning methods available for LLMs (Brown et al., 2020).

To ensure a fair few-shot setup, we randomly extracted one hate and one non-hate sample from the original training dataset for each evaluation dataset and fed them as the few-shot samples. We used a randomly sampled subset as test data, similar to previous studies (Roy et al., 2023; Zhang et al., 2024), to maintain the same number of positive and negative samples and to overcome the cost of LLMs. Finally, we used an IHC and SBIC test dataset of 1K with equal sample proportions.

In all experiments with the four LLMs, few-shot learning consistently outperforms zero-shot learning. On Llama3-8B and Claude3-Haiku, which are known to be relatively lightweight models, zero-shot exhibited F1-scores ranging between 15.83 and 37.62, but after a few-shot sample was injected, performance increased to between 67.77 and 73.22. This suggests that the few-shot prompting strategy is effective for implicit hate speech detection.

From a cost perspective, LLMs still require a large amount of resources and have lower performance compared to fine-tuned language models that have been trained on the specific dataset.

Therefore, we argue that our method of training specialized detection models at a relatively low cost still has advantages over in-context learning with LLMs.