LIDA: Lexical-Based Imbalanced Data Augmentation for Content Moderation

Anonymous PACLIC submission

Abstract

Data augmentation (DA) has attracted considerable attention as an alternative for collecting more data without additional human annotation efforts, particularly in low-resource, sensitive, and class-imbalanced tasks. However, the majority of current approaches are designed for the general domain with often balanced data, while in specific tasks like content moderation, the data is often with a skewed distribution. The situation is further exacerbated by data sensitivity, making it unlikely or costly to obtain additional human annotations. To fill this research gap, our paper presents a lexical-based imbalanced data augmentation (LIDA) approach for content moderation. LIDA is an easy-to-implement and explainable DA method that utilizes sensitive lexicons and randomly inserts sensitive lexicons into negative samples for converting them into positive ones. In this way, LIDA can achieve a balanced dataset for avoiding skewed distribution problems. We validate our model on two datasets, namely Wiki-TOX and Wiki-ATT, to show the superior performance of our proposed algorithm compared to other rule-based data augmentation baselines, and p-values are presented to demonstrate its effectiveness and stability.

1 Introduction

Cyberbullying and harassment have become significant concerns, as they have a negative impact on users who are exposed to inappropriate user-generated content in various forms, such as violent, disturbing, depressive, or fraudulent materials. These experiences can ultimately lead to detrimental effects on their mental health (Patel et al., 2007; Sedgwick et al., 2019). Hence, content moderation holds both significant business and research value for online mental and social communities (McManus et al., 2016).

With the ever-growing volume of online content, automatic moderation has emerged as a promising approach for content moderation, essentially serving as a subtask within text classification (Matamoros-Fernández and Farkas, 2021). Similar to other text classification tasks (Xiang et al., 2021), the effectiveness of content moderation largely depends on the quality and quantity of training data. However, content moderation is typically domain-specific, which is challenging in creating a gold-standard dataset that requires considerable domain expertise and resources. To overcome this challenge, data augmentation (DA) has been proposed as a solution to increase the diversity and quantity of training data without the need for additional data collection or annotation. The data augmentation approach has the potential to enhance the performance and generalizability of content moderation models (Feng et al., 2021).

To our knowledge, existing efforts on DA mostly focus on the general-domain text classification tasks (Karimi et al., 2021; Ren et al., 2021; Xiang et al., 2021; Yoo et al., 2021). Studies of DA in content moderation are lacking, especially for the moderation related to toxic and abusive messages (Ibrahim et al., 2018). Unlike general-domain text classification tasks such as sentiment analysis, these tasks are often challenged by a skewed data distribution among different categories. For example, prior studies suggested that the average proportion of examples in the negative category (e.g., the contents following Twitter community rules without hate/racism/sexism information) among a sample of seven Twitter datasets was over 80% (Zhang and Luo, 2019).

In content moderation, techniques based on rules and lexical features have been widely used (Feng et al., 2021). However, leveraging lexical features for DA in content moderation is rarely investigated. That leaves a research gap we can incorporate the existing lexicons to augment text data for moderation (Koufakou et al., 2020; Xiang et al., 2021).

This paper proposes an easy-to-implement yet effective DA approach for automatic content mod-

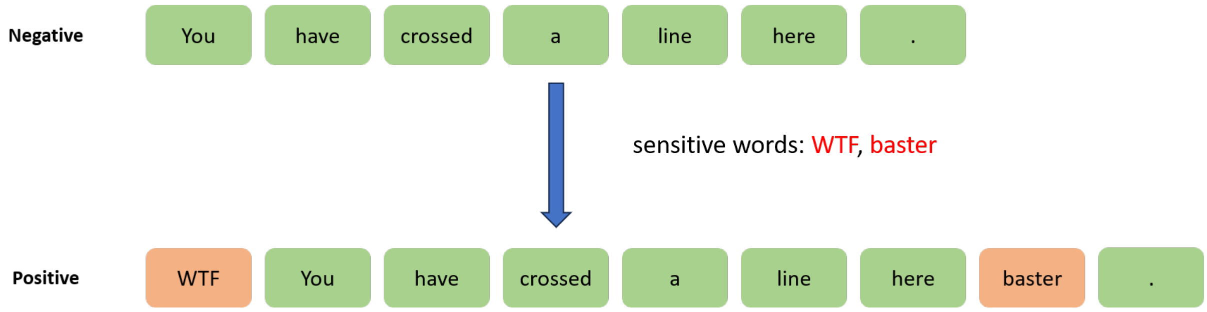

Figure 1: A sample for the process of LIDA. We insert sensitive words, **WTF** and **baster** (orange blocks), into the original negative sample (green blocks) to convert it to positive. Negative samples refer to the samples that pass the moderation, and positive samples mean the samples that fail the moderation.

eration. Different from prior approaches that heavily depend on large word lists (Koufakou et al., 2020), we randomly select sensitive words from a 104-word lexicon collected from Wiktionary [1] and Hatebase [2], and insert the words into the original negative samples to convert them into positive ones. Via leveraging the lexical knowledge and features, we can obtain relatively balanced data without soft labels (Kwon and Lee, 2022; Shorten et al., 2021).

Specifically, as shown in Figure 1, we randomly select two sensitive words, *WTF* and *baster*, from our wordlist and insert them into a negative sample *"You have crossed a line here"*, via which this sentence is converted into positive. Regardless of where we incorporate the lexical features within the original sentence, the resulting sentence would become a positive sample that should be eliminated due to the inability of sensitive words to pass moderation. Intuitively, our proposed LIDA algorithm has the capability to transform negative samples into positive ones for the purpose of data augmentation.

Content Warning. This article contains examples of hateful and abusive language. All examples are taken from Wikitionary [1] and Hatebase [2] to illustrate its composition.

2 Related Work

Current DA methods could be roughly classified into two categories: rule-based data augmentation, and generative model-based data augmentation.

2.1 Rule-Based DA

The rule-based DA approaches typically operate on words, phrases, or sequences in the original data, such as swap, deletion, insertion and replacement. Wei and Zou (2019) proposed the Easy Data Augmentation (EDA) method which provides four operations on a given sentence, 1) randomly selecting n words to be replaced by their synonyms; 2) inserting synonyms of random words in random positions; 3) swapping two words, and 4) deleting a word randomly with probability p. In contrast to EDA that based on word operations, which might change the labels of augmented data, Karimi et al. (2021) introduced An Easier Data Augmentation (AEDA) which only inserts punctuation marks into the original data. Therefore, it preserves class labels invariant. Additionally, building up a learnable and compositional paradigm for DA is another outstanding rule-based mix DA technique, such as Text AutoAugment (TAA) (Ren et al., 2021). Morevoer, Xiang et al. (2021) introduced an approach making use of the part-of-speech (POS) focused lexical substitution for data augmentation (PLSDA). They exploited POS information to identify words to be replaced and investigate different augmentation strategies to find semantically related substitutions based on synonyms on WordNet. Nevertheless, the mentioned rule-based methods cannot obtain balanced datasets via data augmentation. Consequently, these DA approaches are limited in handling imbalanced data.

To overcome the limitations of imbalanced data, our proposed method obtains fairly balanced data by inserting lexical features into the raw negative data to convert negative cases into positive ones. Meanwhile, our approach does not depend on soft label predictions.

2.2 Generative-Based DA

Generative-based DA approaches usually employ large-scale language models (LLMs) to synthesize

[1] https://en.wiktionary.org/wiki/Category:English_swear_words

[2] https://hatebase.org/

new augmented samples based on the original data (Anaby-Tavor et al., 2020; Yoo et al., 2021; Dai et al., 2023; Bayer et al., 2023; Xie et al., 2017). Anaby-Tavor et al. (2020) built a data augmentation pipeline based on generative pre-training (GPT) (Radford et al., 2018) with limited labeled data, and then filtered the augmented data on a classifier trained on the original data. Using a similar GPT-based model, Yoo et al. (2021) mixed real samples to synthesize realistic text samples via GPT-3 (Brown et al., 2020), and leveraged textual perturbations and knowledge distillation from pre-trained transformer-based language models to predict soft-labels. Inspired by the recent success of ChatGPT, which demonstrated improved language comprehension abilities, Dai et al. (2023) proposed a text data augmentation approach based on ChatGPT (named AugGPT). AugGPT rephrases each sentence in the training samples into multiple conceptually similar but semantically different samples. The augmented samples can then be used in downstream model training.

Although generative DA methods have the advantage of synthesizing diverse and fluent augmented samples, they tend to suffer from the high cost of pre-training and inference. Most importantly these approaches would heavily rely on predicted soft labels for data augmentation.

From rule-based manipulations to generative models, an ideal DA strategy should be as simple as to implement while being able to boost model performance, given the purpose of DA is to provide alternatives for gathering additional data. Most studies trade off between the two of these (Feng et al., 2021).

3 Methodology

3.1 Lexical Features

Firstly, we randomly collect English swear words from Wikitionary [1] and hate words from Hatebase [2], which is a collaborative, regionalized repository of multilingual hate speech, developed in partnership between the Dark Data Project [3] and The Sentinel Project [4]. We collect a total of 140 words in this step. Secondly, two doctoral students who are native speakers of English review and filter the wordlist. Since the ambiguity of sensitive words would affect the performance of our model (e.g., **northern monkey**). We discuss and analyze the lexical ambiguity and insertion strategy in Section 5.4. Finally, we get 104 sensitive words (see Appendix B for the detailed word list).

3.2 LIDA Method

We present LIDA method in Algorithm 1. In our model, a raw training sentence is represented as $s = [w_1, w_2, ..., w_i, ..., w_l]$, where w_i is the i^{th} word or token and l is the length of a sentence. Supposing that there are M training samples, including numbers of N negative samples and P positive samples, we then set up augmentation proportion t as a hyperparameter, in order to avoid augmenting with too much noise. Utilizing data noise could be an effective technique for DA since operations like insertion have the potential to disrupt the original sentence order, leading to information loss, the introduction of noise, and even label changes (Kumar et al., 2020). For example, Karimi et al. (2021) used a method that either by replacing words selected from the uni-gram frequency distribution or by inserting the underscore character as a placeholder. However, adding too much noise could mislead the model and affect the performance. Thereby, augmentation proportion t is a crucial hyper-parameter of our proposed algorithm.

For each loop, we randomly generate d of 1, 2, or 3 as the number of lexical features $[LF]$ selected from the lexicon. The operation can be written as:

$$[LF]^{d=1,2,3} = select(d). \quad (1)$$

Parameter d plays a significant role in LIDA (as shown in Section 5.4), to avoid word ambiguities. Finally, $[LF]^{d=1,2,3}$ is inserted into a negative sentence s to return a positive sentence s':

$$s' = insert([LF]^{d=1,2,3}, s) \quad (2)$$

We combine the augmented data s' and original data s as the new training set. Since sensitive words violate the policies of the content moderation community, it is reasonable to assume that negative samples would be converted into positive ones after adding these words.

3.3 Content Moderation Pipeline

Following data augmentation, we acquire a balanced dataset. Subsequently, both the augmented and original data can be utilized for downstream model training purposes. Specifically, we show the pipeline for content moderation using BERT

[3] https://darkdataproject.org/

[4] https://thesentinelproject.org/

Algorithm 1: LIDA Algorithm

```
Input: Number of training samples $M$, including number of negative samples $N$ and positive
       samples $P$.
Output: augmented positive sample $s'$. Totally,number of negative samples $N$ and positive
        samples $P + \lfloor N * t \rfloor$
Initialize augmentation proportion $t$
for $i = 1$ to $P + \lfloor N * t \rfloor$ do
    d = 1 or 2, or 3
    $[LF]^{d=1,2,3} = select(d)$
    $s' = insert([LF]^{d=1,2,3}, s)$
end
Return $s'$
```

with LIDA. Firstly, we leverage LIDA for data augmentation to obtain balanced augmented data. To perform content moderation using BERT, the process involves data preprocessing, where the text is tokenized and converted into BERT input format. Next, a BERT model is built by adding additional layers on top of the pre-trained BERT model, including a pooling layer and fully connected layers for classification. We fine-tune the model using the augmented data, optimizing it with a chosen loss function and optimizer. Finally, the trained model can be used for inference by preprocessing new text data and passing it through the model to obtain predictions for content moderation. Our model can apply to all mainstream classification methods.

4 Experiments

We introduce our benchmark datasets and selected baselines in this section. The experiment settings are shown in Appendix A.

4.1 Benchmark Datasets

We conduct experiments on two public datasets, Wikipedia Toxic [5] and Wikipedia Personal Attack datasets [6] from Wikipedia Talk dataset (Wulczyn et al., 2017). These datasets contain human annotations for toxic and personal attack behaviour.

We model the tasks in both datasets as binary classification, named after Wiki-TOX and Wiki-ATT. Table 1 shows the basic information of the two datasets. Compared with (Karimi et al., 2021; Wei and Zou, 2019) whose experiment used balanced datasets, the positive samples in these two datasets constitute approximately 10.1% and 7.5% individually.

Wiki-ATT. The Wikipedia Personal Attacks dataset (Wulczyn et al., 2017) is a subset of the Wikipedia Comment Corpus, containing 63M comments from English discussion pages and articles during 2004-2015. Each comment was annotated and identified as personal attacks by at least 10 workers. The dataset comprises five classes: quoting attack, recipient attack, third-party attack, other attack, and no attack. While quoting, recipient, third-party, and other attacks are different types of attacks, they are still considered attacks. Therefore, we combine the four categories as positive and consider the no-attack category as negative.

Wiki-TOX. Wulczyn et al. (Wulczyn et al., 2017) presented the toxic comment dataset, which is widely used for toxic detection (Bodapati et al.) and Kaggle competition founded by Jigsaw and Google. The dataset comprises seven classes of comments, namely toxic, severe toxic, obscene, threat, insult, and identity hate. Among these classes, the six types of comments displaying any form of toxicity are regarded as positive samples, whereas comments lacking any such characteristics are categorized as negative.

4.2 Selected Baselines

We selected three groups of baselines to validate our proposed LIDA method.

In the first group of baseline, we compare LIDA with three neural networks with no augmentation, which are Convolutional Neural Network (CNN) (Kim, 2014), Recurrent Neural Network (RNN) (Liu et al., 2016) and BERT (base, cased version) (Devlin et al., 2019). Glove vectors (300 dimensions)(Pennington et al., 2014) are used as pre-trained weights for the embedding layer in CNN.

[5] https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data

[6] https://www.kaggle.com/datasets/jigsaw-team/wikipedia-talk-labels-personal-attacks

	Data			Train			Val			Test		
	Total	**N**	**P**	**Total**	**N**	**P**	**Total**	**N**	**P**	**Total**	**N**	**P**
Wiki-TOX	159,495	143,346	16,149	139,495	125,413	14,082	10,000	8,966	1,034	10,000	8,967	1,033
WIKI-ATT	115,864	107,190	8,674	95,864	88,762	7,102	10,000	9,218	782	10,000	9,210	790

Table 1: The statistics of Wiki-TOX and Wiki-ATT. Train, Val and Test represent the training set, validation set and test set respectively. N and P refer to the number of negative and positive samples.

For RNN, we choose a simple LSTM model, which consists of one layer with 128 hidden units and random initialization weights for the embedding layer.

In the second group of baselines, we compare LIDA with several related rule-based methods and generative-based text data augmentation methods that have been published in recent peer reviewed conferences and journals.

- **Easy Data Augmentation (EDA)** (Wei and Zou, 2019). Given a training sample, EDA randomly employs four operations: 1)randomly selecting n words to be replaced by their synonyms; 2) choosing a synonym of a random word and inserting it in a random position, repeating it n times; 3) swapping the positions of two words, repeating it n times; and 4) randomly deleting a word with probability p. According to the recommendations presented in the paper, we set up the proportion of words to be edited to 0.05 [7].

- **An Easier Data Augmentation (AEDA)** (Karimi et al., 2021) is a method that offers a simpler approach to data augmentation by randomly inserting punctuations into the original text. Since rule-based approaches rely on the hyperparameters (Ren et al., 2021), we set the punctuation ratio to 0.3 based on the implementation of AEDA [8].

Additionally, our group 3 baselines include the recently popular used GPT3 model for data augmentation. **GPT3Mix** (Yoo et al., 2021) mixes real samples to synthesize realistic text samples via GPT3 (Brown et al., 2020), and leverages textual perturbations and knowledge distillation from pre-trained transformer-based language models to predict soft labels.

5 Results and Analysis

In this section, we show the main results and compare to original DNN models, ruled-based and generative-based baselines in Section 5.1, 5.2, 5.3, respectively. In Section 5.4, we conduct two groups of ablation experiments for discussing and analysing the effect of augmentation proportion and insertion strategy of our proposed LIDA method.

5.1 Compare to original DNN models

LIDA and baselines are evaluated both on Wiki-TOX and Wiki-ATT. The overall performance is measured by F1-score and AUC. The statistical significance and stability of the experimental outcomes are ensured by employing the p-value testing approach.

Compared with the vanilla models without data augmentation, Table 2 shows that LIDA gives a performance boost in F1-score and AUC for all models on both two datasets. On average, LIDA gets 4.62, 2.55, and 6.13 F1-score improvement on the three models, respectively. Significantly, the p-values are less than 0.05 for the three models, indicating that models trained with our method significantly outperform those without augmentation.

5.2 Compare to Rule-Based Baselines

Since LIDA technique is inherently a rule-based DA technique, it is particularly important to compare it with other rule-based methods. Table 3 shows that LIDA outperforms EDA on average by 3.40 (F1-score) and 2.67 (AUC) on Wiki-TOX and by 2.67 (F1-score) and 1.40 (AUC) on Wiki-ATT. Similarly, compared with AEDA, our algorithm demonstrates an average improvement of 1.17 (F1-score) and 2.66 (AUC) on Wiki-TOX and 2.15 (F1-score) and 1.10 (AUC) on Wiki-ATT. Table 4 reports the P-values of LIDA against EDA and AEDA on three models, and statistically confirms that our method outperforms the two rule-based approaches (all values are less than 0.05). Notably, the results are particularly promising for content moderation, as the task requires the ability to de-

[7] https://github.com/jasonwei20/eda_nlp

[8] https://github.com/akkarimi/aeda_nlp/blob/master/code/aeda.py

Datasets	Models	No Aug		LIDA		Improvement		P-Value	
		F1	AUC	F1	AUC	F1	AUC	F1	AUC
Wiki-TOX	CNN	65.67	75.99	72.38	82.11	6.71	6.12	0.0000	0.0000
	RNN	36.84	61.26	38.52	66.06	1.68	4.80	0.0000	0.0000
	BERT	75.59	88.95	83.65	91.57	8.06	2.62	0.0000	0.0082
Wiki-ATT	CNN	69.02	79.26	71.54	81.44	2.52	2.18	0.0066	0.0175
	RNN	43.30	64.16	46.51	67.10	3.21	2.94	0.0000	0.0000
	BERT	76.82	89.04	81.02	91.20	4.20	2.16	0.0001	0.0051
Average	CNN	67.34	77.62	71.96	81.77	**4.62**	**4.15**	-	-
	RNN	40.07	62.71	42.52	66.58	**2.45**	**3.87**	-	-
	BERT	76.20	88.99	82.33	91.38	**6.13**	**2.39**	-	-

Table 2: Compare LIDA to no augmentation. All experiments have been conducted 10 times and we obtained the averages of 10 times as results. "-" denotes the case where no results are available.

tect as much harmful content as possible and avoid false negatives.

We also observe that the performance improvement of our algorithm on RNN is lower than that on CNN and BERT. This observation may be attributed to the simplicity of the RNN model used in our study (see Section 4.2), which lacks the pre-trained vectors necessary for leveraging word information. In light of the structure of the RNN model we selected, the LIDA technique does not result in a significant improvement in the model's performance compared to CNN and BERT. This finding suggests that the effectiveness of DA techniques in improving model performance is closely associated with the model's structure and complexity, which is also reflected in the significant impact of pre-trained word vectors on performance.

5.3 Compare to Generative-Based Baseline

To assess the effectiveness and efficiency of our proposed rule-based data augmentation algorithm, we conduct comparisons not only with other rule-based baselines, but also with a state-of-the-art generative-based model, GPT3Mix (Yoo et al., 2021). Since GPT3Mix has the capability to synthesize diverse and fluent augmented samples owing to the power of large-scale pre-trained language models, it shows a significant improvement in performance over LIDA on CNN, RNN, and BERT, as discussed in Related Work (Section 2) and presented in Table 5.

However, GPT3Mix suffers from the heavy cost of pre-training and fine-tuning. GPT-based models commonly require computing resources and consume more time to fine-tune for downstream tasks. For example, GPT-3 has 175 billion parameters trained on 45 TB corpus (Brown et al., 2020). Even fine-tuning GPT3Mix requires considerably more time than EDA, AEDA, and LIDA.

Models	Wiki-TOX		Wiki-ATT	
	F1	AUC	F1	AUC
CNN	65.67	75.99	69.02	79.26
+EDA	68.08	78.04	69.17	79.71
+AEDA	71.12	79.66	69.41	79.89
+LIDA	**72.38**	**82.11**	**71.54**	**81.44**
RNN	36.84	61.26	43.30	64.16
+EDA	37.04	61.40	44.38	65.45
+AEDA	37.86	61.83	45.24	65.91
+LIDA	**38.52**	**66.06**	**46.51**	**67.10**
BERT	75.59	88.95	76.82	89.04
+EDA	79.22	89.62	77.51	90.39
+AEDA	82.05	90.28	77.97	90.64
+LIDA	**83.65**	**91.57**	**81.02**	**91.20**

Table 3: Compare to rule-based baselines. overall performance is measured by F1 and AUC. F1: F1-score, AUC: Area Under the Receiver Operating Characteristics (AUC-ROC). All experiments have been conducted 10 times and we got the averages of 10 times as results.

Therefore, it is meaningful and feasible to sacrifice a small portion of performance in exchange for a more efficient and economical DA method.

5.4 Ablation studies

Effect of Augmentation Proportion. The impact of augmentation proportion is analyzed in the con-

		Wiki-TOX		Wiki-ATT	
		F1	**AUC**	**F1**	**AUC**
CNN	EDA	0.0000	0.0000	0.0127	0.0713
	AEDA	0.0398	0.0055	0.0455	0.0376
RNN	EDA	0.0004	0.0000	0.0031	0.0022
	AEDA	0.0331	0.0000	0.0277	0.0075
BERT	EDA	0.0000	0.0017	0.0024	0.0087
	AEDA	0.0018	0.0192	0.0013	0.0206

Table 4: P-values are presented between LIDA with EDA and AEDA.We run the experiments in 10 times. They can demonstrate the performance of our algorithm statistically.

Datasets	**Models**	**GPT3Mix**		**LIDA**		**Improvement**	
		F1	**AUC**	**F1**	**AUC**	**F1**	**AUC**
Wiki-TOX	CNN	74.48	83.98	72.38	82.11	-2.10	-1.87
	RNN	38.94	66.39	38.52	66.06	-0.42	-0.33
	BERT	86.32	94.53	83.65	91.57	-2.67	-2.96
Wiki-ATT	CNN	72.94	83.30	71.54	81.44	-1.40	-1.86
	RNN	47.03	67.73	46.51	67.10	-0.52	-0.63
	BERT	84.28	93.11	81.02	91.20	-3.26	-1.91
Average	CNN	73.71	83.64	71.96	81.77	**-1.75**	**-1.87**
	RNN	42.99	67.06	42.52	66.58	**-0.47**	**-0.48**
	BERT	85.30	93.82	82.33	91.38	**-2.97**	**-2.44**

Table 5: A Comparison between the Performance of LIDA and GPT3Mix on Wiki-TOX and Wiki-ATT.

text of Wiki-TOX for CNN and LSTM models. It is noted that the augmentation ratio is a critical parameter, given insertion strategies and other parameters. Figure 2 illustrates that the optimal augmentation ratio is typically within the [50,60] interval. While an exact interval for augmentation proportion has yet to be determined, the findings demonstrate that the augmentation ratio is a critical hyperparameter.

Effect of Insertion Strategy. We conduct a comparison of three insertion strategies, specifically, $d = 1$, $d = 2$, and $d = 1, 2, 3$. Our findings, presented in Table 6, indicate that different strategies significantly impact the models' performance. Notably, when $d = 1$, the models achieve the poorest results. One possible explanation for this outcome is the presence of ambiguous words, such as "**northern monkey**". It is used in the south of England as a slang word, relating to the supposed stupidity and lack of sophistication of those in the north of the country. In some cases, this has been adopted in the north of England, with a pub in Leeds even taking the name "The Northern Monkey". When used to attack northerners, **northern monkey** is a hate word, which can be inserted into the raw sentence to convert it from negative to positive. However, in a general context, it also means the monkeys which live in the north. To mitigate this issue, we propose a random combination insertion strategy that can help reduce the influence of the ambiguity at a phrase-level.

6 Conclusion

In this paper, we present a simple but effective DA method called lexical-based imbalanced data augmentation (LIDA) for content moderation. LIDA leverages lexical features to transform negative samples into positive samples, thereby obtaining balanced data without soft labels or human annotation. Experiments show that LIDA can substantially improve the generalization ability of models as well as alleviate a burden of human annotation. We evaluate our model on benchmark moderation tasks. The results show our algorithm outperforms

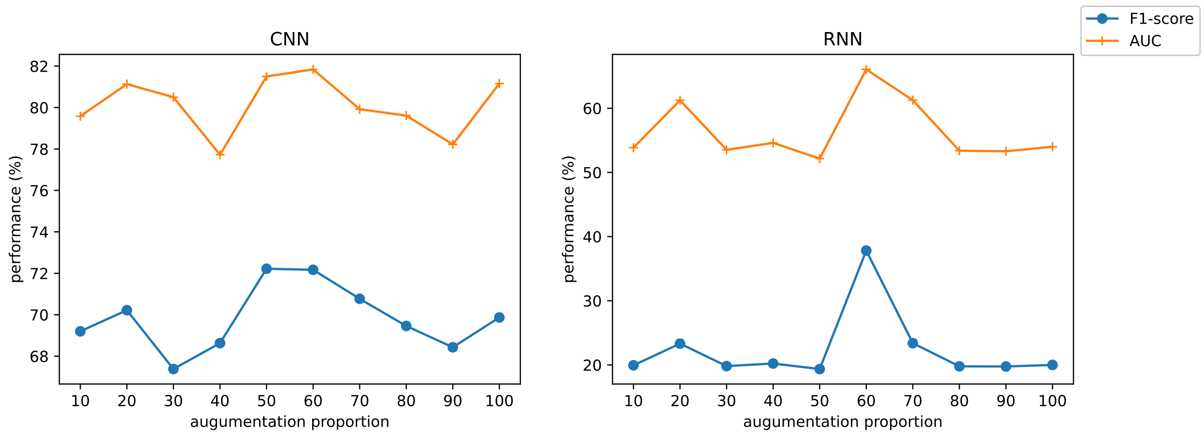

Figure 2: Explore augmentation proportion on Wiki-TOX dataset.

	CNN		RNN		BERT	
	F1	**AUC**	**F1**	**AUC**	**F1**	**AUC**
d=1	65.79	79.79	37.49	65.62	80.81	86.22
d=2	70.81	81.22	37.36	**66.09**	81.89	88.33
d=1, 2, 3	**72.38**	**82.11**	**38.52**	66.06	**83.65**	**91.57**

Table 6: Results of different insertion strategies on Wiki-TOX. *d* denotes the number of lexical features that are inserted into the original sample.

other rule-based baselines, and the statistical analysis with p-values indicates the effectiveness and stability of the LIDA method. Thus, our method can be a competitive alternative to the rule-based solution for augmenting imbalanced data. Although, our model shows inferior performances compared with the generative-based DA methods based on large-scale language models, considering the cost of computational resources, explain-ability issues, and data privacy problems, the rule-based methods like LIDA can still find its position in automatic moderation given its low computational cost, high performance, and the ability to leverage human moderation knowledge.

7 Limitations

Although our proposed algorithm outperforms rule-based data augmentation algorithms EDA and ADEA, this study has some limitations as below:

- The utilization of large-scale pre-trained language models endows GPT3Mix with the capability of generating a vast array of fluent and diverse augmented samples, leading to superior performance in comparison to our proposed method. Nevertheless, it is noteworthy that GPT3Mix incurs a substantial computational burden due to the intensive nature of its pre-training and fine-tuning processes.

- The findings demonstrate that the augmentation ratio is a critical hyperparameter, with LIDA being sensitive to it. However, an exact interval for augmentation proportion has yet to be determined.

- Moreover, as mentioned in Section 5.4, lexical features such as "northern monkey" significantly affect the performance of our proposed method. The performance of LIDA is influenced by the choice of lexicons and the corresponding insertion strategy. However, it is noted that using appropriate lexicons has the potential to enhance the performance even further.

- Currently, our proposed algorithm can be used for binary classification tasks only. For multi-classification tasks, we need to collect and create a multi-category sensitive lexicon, e.g., toxic, obscene, threat, insult and identity hate. And then using LIDA to insert these sensitive words into the corresponding labelled sentence for data augmentation.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Markus Bayer, Marc-André Kaufhold, Björn Buchhold, Marcel Keller, Jörg Dallmeyer, and Christian Reuter. 2023. Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. *International journal of machine learning and cybernetics*, 14(1):135–150.

Sravan Bodapati, Spandana Gella, Kasturi Bhattacharjee, and Yaser Al-Onaizan. Neural word decomposition models for abusive language detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 135–145.

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.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, et al. 2023. Chataug: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988.

Mai Ibrahim, Marwan Torki, and Nagwa El-Makky. 2018. Imbalanced toxic comments classification using data augmentation and deep learning. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 875–878. IEEE.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Anna Koufakou, Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2020. Hurtbert: Incorporating lexical features with bert for the detection of abusive language. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 34–43.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26.

Soonki Kwon and Younghoon Lee. 2022. Explainability-based mix-up approach for text data augmentation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2873–2879.

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.

Sally McManus, Paul E Bebbington, Rachel Jenkins, and Terry Brugha. 2016. *Mental health and wellbeing in England: the adult psychiatric morbidity survey 2014*. NHS digital.

Vikram Patel, Alan J Flisher, Sarah Hetrick, and Patrick McGorry. 2007. Mental health of young people: a global public-health challenge. *The Lancet*, 369(9569):1302–1313.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Shuhuai Ren, Jinchao Zhang, Lei Li, Xu Sun, and Jie Zhou. 2021. Text autoaugment: Learning compositional augmentation policy for text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9029–9043.

Rosemary Sedgwick, Sophie Epstein, Rina Dutta, and Dennis Ougrin. 2019. Social media, internet use and suicide attempts in adolescents. *Current opinion in psychiatry*, 32(6):534.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8(1):1–34.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web*, pages 1391–1399.

Rong Xiang, Emmanuele Chersoni, Qin Lu, Chu-Ren Huang, Wenjie Li, and Yunfei Long. 2021. Lexical data augmentation for sentiment analysis. *Journal of the Association for Information Science and Technology*, 72(11):1432–1447.

Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. 2017. Data noising as smoothing in neural network language models. In *In 5th International Conference on Learning Representations, ICLR*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239.

Ziqi Zhang and Lei Luo. 2019. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, 10(5):925–945.

A Experiment Settings

We adopt the Adam optimizer (Kingma and Ba, 2014) along with a linear learning rate scheduler with a warm-up ratio of 0.05. The experiments are conducted on RTX 6000 GPU (24G Memory) and GTX 3090 GPU (24G Memory). For each experimental task, each model run 10 times and the average values are taken as the result. Moreover, the p-values have been considered to prove and verify the reliability and stability of the experimental results.

B Sensitive Wordlist

ID	Sources	Words
001	Wikitionary	*arse*
002	Wikitionary	*ass*
003	Wikitionary	*asshole*
004	Wikitionary	*bastard*
005	Wikitionary	*bitch*
006	Wikitionary	*bollocks*
007	Wikitionary	*brotherfucker*
008	Wikitionary	*bugger*
009	Wikitionary	*bullshit*
010	Wikitionary	*child-fucker*
011	Wikitionary	*Christ on a bike*
012	Wikitionary	*Christ on a cracker*
013	Wikitionary	*cocksucker*
014	Wikitionary	*crap*
015	Wikitionary	*cunt*
016	Wikitionary	*damn*
017	Wikitionary	*effing*
018	Wikitionary	*fatherfucker*
019	Wikitionary	*frigger*
020	Wikitionary	*fuck*
021	Wikitionary	*goddamn*
022	Wikitionary	*godsdamn*
023	Wikitionary	*hell*
024	Wikitionary	*holy shit*
025	Wikitionary	*horseshit*
026	Wikitionary	*in shit*
027	Wikitionary	*Jesus Christ*
028	Wikitionary	*Jesus fuck*
029	Wikitionary	*Jesus H. Christ*
030	Wikitionary	*Jesus Harold Christ*
031	Wikitionary	*Jesus wept*
032	Wikitionary	*Jesus, Mary and Joseph*
033	Wikitionary	*Judas Priest*
034	Wikitionary	*motherfucker*
035	Wikitionary	*nigga*
036	Wikitionary	*piss*
037	Wikitionary	*prick*
038	Wikitionary	*shit*
039	Wikitionary	*shit ass*
040	Wikitionary	*sisterfucker*
041	Wikitionary	*slut*
042	Wikitionary	*son of a bitch*
043	Wikitionary	*son of a whore*
044	Wikitionary	*sweet Jesus*
045	Wikitionary	*twat*
046	Hatebase	*buttfucker*
047	Hatebase	*assplay*
048	Hatebase	*sucker*
049	Hatebase	*homophobic slurs*
050	Hatebase	*nerdiness*
051	Hatebase	*putz*
052	Hatebase	*ass-rape*
053	Hatebase	*ponce*

054 Hatebase *narcism*

055 Hatebase *muthafucker*

056 Hatebase *dastardliness*

057 Hatebase *african-negros*

058 Hatebase *virgin*

059 Hatebase *arsehole*

060 Hatebase *crook*

061 Hatebase *self-destruction*

062 Hatebase *self-annihilation*

063 Hatebase *vestal*

064 Hatebase *pervert*

065 Hatebase *self harm*

066 Hatebase *slay*

067 Hatebase *felon*

068 Hatebase *virgo the virgin*

069 Hatebase *outrage*

070 Hatebase *self injury*

071 Hatebase *shoot down*

072 Hatebase *whoreson*

073 Hatebase *ill-treat*

074 Hatebase *terrorist*

075 Hatebase *bastard*

076 Hatebase *blackguard*

077 Hatebase *maltreat*

078 Hatebase *ill-usage*

079 Hatebase *mistreat*

080 Hatebase *suicide*

081 Hatebase *dickhead*

082 Hatebase *maltreatment*

083 Hatebase *virginal*

084 Hatebase *prick*

085 Hatebase *shit*

086 Hatebase *ravish*

087 Hatebase *rape*

088 Hatebase *ill-use*

089 Hatebase *slaying*

090 Hatebase *sexually assault*

091 Hatebase *violate*

092 Hatebase *cocksucker*

093 Hatebase *wtf*

094 Hatebase *self loathe*

095 Hatebase *gay*

096 Hatebase *lesbian*

097 Hatebase *terrorist*

098 Hatebase *murder*

099 Hatebase *assault*

100 Hatebase *kill*

101 Hatebase *robbery*

102 Hatebase *dumbcunt*

103 Hatebase *topless*

104 Hatebase *dickdipper*