Comparison of Data Augmentation Techniques on the HateXplain Dataset

Evan Chan,[1] Kieran Berton,[1] and Chukwudera Mojekwu[1]
[1]*School of Information, University of California, Berkeley, CA 94720,*
9 April 2022

We tested six different training set data augmentation techniques at various levels of augmentation on the HateXplain dataset presented in Mathew et al., 2020. Augmented models were evaluated against the baseline with respect to model performance (Accuracy, macro F1 Score and ROC-AUC) and bias metrics. Data augmentation was shown to improve model performance at lower levels of augmentation by 1-2% accuracy. Higher levels of augmentation produced overfitted models. Contextual-word-based techniques were also shown to improve or have no effect on model bias. We conclude that data augmentation is a viable tool for improving hate speech detection models.

Disclaimer: Some of the content represented here contains strong hateful or offensive language that may be considered stereotypical or prejudiced. This language does not reflect the opinions or values of the authors and readers should proceed with caution.

INTRODUCTION

Identifying online hate speech is a uniquely difficult task due to the differences in what can be classified as 'hate speech' in different contexts or domains (Schmidt and Weigan 2017). While there is no formal definition, hate speech is commonly defined as language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group (Davidson et. al 2017). It is therefore important to ensure that models for automated hate speech detection are robust to various nuances in meaning that may be unique to the domain of discourse. Good automated models would therefore be capable of high prediction accuracy while at the same time minimizing bias towards certain affected groups.

Towards this goal, Mathew, et al. created the HateXplain dataset, which, among other contributions, enables the calculation of metrics on model performance, bias, and explainability (Mathew et al., 2020). However, HateXplain relies on human annotators to label various aspects of each speech example which is costly and time consuming. To maximize the value of human annotated data, we are interested in testing whether various data augmentation techniques will improve the model's ability to distinguish between these various nuances in meaning, resulting in improved model performance in all three areas.

While the HateXplain paper examined the performance of several different classification models, we will focus on BERT for this study. BERT-based models performed best in terms of both classification accuracy and bias, while also performing well in some measures of bias and explainability. Previous studies have shown that data augmentation can be effective in improving classification performance with BERT (Kumar et. al 2019). These studies also noted that there is a diminishing return as the training set size increases, we will therefore also be testing various levels of augmentation.

BACKGROUND AND METHODS

The HateXplain Data Set

HateXplain (Mathew et al., 2020) is the first benchmark dataset covering multiple aspects of online offensive and toxic hate speech. It consists of 20,148 social media posts taken from Twitter and Gab. Each post in the dataset is annotated by three different humans from three different perspectives: a 3-class classification (i.e., hate, offensive or normal), the target community (i.e., the community that has been the victim of hate speech/offensive speech in the post), and the rationales, i.e., the portions of the post on which their labeling decision (as hate, offensive or normal) is based. Together these allow the evaluation of a prediction model's performance, bias, and explainability. For this study, we will focus on model performance and bias.

Though the human annotators were given specific definitions, as above, on what constitutes hate speech, offensive, and normal posts. Their classification is inherently subjective, therefore the final classification and target groups are determined by majority vote. This reduces the dataset to 19,229 posts with sufficient agreement from the annotators. Historically, datasets used to train hate speech classification models have been unbalanced towards normal posts due to the relative difficulty in obtaining examples of hate speech from "mainstream" social media networks (Cao and Lee 2020). The creators of HateXplain added posts from Gab to largely solve this issue, resulting in a relatively well balanced dataset

	Twitter	Gab	Total	%
Hateful	708	5,227	5,935	29%
Offensive	2,328	3,152	5,480	27%
Normal	5,770	2,044	7,814	39%
Undecided	249	670	919	5%
Total	9,055	11,093	20,148	

TABLE I: HateXplain Dataset Class Distribution

Target Groups	Categories
Race	African, Arabs, Asians Caucasian, Hispanic
Religion	Buddhism, Christian, Hindu, Islam, Jewish
Gender	Men, Women
Sexual Orientation	Heterosexual, LGBTQ
Miscellaneous	Indigenous, Refugee/Immigrant

TABLE II: Identified Target Groups

across all three classes (Mathew et al., 2019). The class distribution is shown in Table 1.

For bias calculations, the authors identified the following groups, shown in Table 2

Model Performance Metrics

Since our class imbalance is minor we can default to the simplest metrics to assess model performance. We will therefore use accuracy as our primary metric. To be consistent with the original HateXplain paper, we will also report Macro F1 and ROC AUC scores.

Data Augmentation

A good portion of improving from our baseline results depends on the size and quality of the data we use to train the models. Acquiring more high quality training examples would be expensive and time consuming to collect, clean and annotate. Learning from the field of computer vision, we decided on data augmentation as an appropriate alternative since it can help train more robust models with smaller training sets (Perez and Wang, 2017). The idea being that inducing some amount of noise into the dataset can be extremely helpful for training a robust model which is important in our case working with real world social media data. To improve on our baseline models, we tested out augmented versions of the original data set using two augmentation techniques: Easy Data Augmentation (EDA) and NLP AUG.

Easy Data Augmentation (EDA)

Wei and Zou (2019) demonstrated EDA's ability to boost performance on text classification tasks. We selected EDA for our dataset because it performed well in preventing overfitting and helping train more robust models. EDA consists of 4 simple operations:

- Synonym Replacement (SR): Randomly choose n words from the sentence that are not stop words. Replace each of these words with one of its synonyms chosen at random.
- Random Insertion (RI): Find a random synonym of a random word in the sentence that is not a stop word. Insert that synonym into a random position in the sentence. Do this n times.
- Random Swap (RS): Randomly choose two words in the sentence and swap their positions. Do this n times.
- Random Deletion (RD): Randomly remove each word in the sentence with probability p.

The vocabulary of words used for random insertion and synonym replacement come from WordNet. For synonym replacement, possible synonyms are found using cosine similarity. These operations were used in various combinations to generate new augmented examples. A probability parameter (between 0 and 1) was used to control the amount of each operation. To isolate the effect of each method, we set the probability at 0.7 for the primary method indicated and 0.1 for all others, this appeared to produce reasonable new examples.

NLP AUG

NLP AUG (Ma, 2019) is a tool for performing data augmentation for NLP machine learning tasks. It supports a wide variety of methods on everything from character to sentence level manipulations. For this study, we will focus on its contextual word insertion and substitution features. The library supports word insertion and substitution with several sets of word embeddings, but we chose BERT to be consistent with our primary classification model.

For both insertion and substitution, a list synonym words is determined by cosine similarity from which a word is chosen randomly. A hyperparameter (aug_p) controls the probability that each word in the piece of text to be augmented receives a substitution or insertion. We set this parameter by manual verifying that the augmented text is still understandable

and arrived at a value of 0.2. This parameter can be further explored at a later date.

Contextual insertion and substitution has been shown to be effective in improving machine learning models in a variety of tasks (Wang and Yang, 2015), (Maimaiti, et al., 2021). The main mechanism for improving model performance was identified as providing additional contexts for rare words and injecting additional noise into the training set. Table 3, shows an example post from the training dataset with all tested augmentations applied.

Bias

Due to the sometimes ambiguous nature of hateful and offensive language and in particular language referring to minority groups, hate speech detection models have the potential to make biased predictions about language referring to those particular groups (Sap et al. 2019). For example, the sentence "I love my ni—s." might be classified as hateful/offensive because of the association of the n-word with the black community, despite the sentence in context not being intentionally hateful. Social media platforms under increasing pressure to respond to online hate speech might risk further suppressing the already marginalized voices of minority groups when using automated methods to remove sensitive content, which would have a negative impact on the target community.

To measure such unintended model bias, we calculated for each of our models and each minority subgroup the AUC based metrics developed by Borkan et al. (2019). These include Subgroup AUC, Background Positive Subgroup Negative (BPSN) AUC, Background Negative Subgroup Positive (BNSP) AUC, and Generalized Mean of Bias AUCs.

- Subgroup AUC: Here, we select toxic and normal posts from the test set that mention the community under consideration. The ROC-AUC score of this set will provide us with the Subgroup AUC for a community. This metric measures the model's ability to separate the toxic and normal comments in the context of the community (e.g., Asians, LGBTQ etc.). A higher value means that the model is doing a good job at distinguishing the toxic and normal posts specific to the community.

- BPSN (Background Positive, Subgroup Negative) AUC: Here, we select normal posts that mention the community and toxic posts that do not mention the community, from the test set. The ROC-AUC score of this set will provide us with the BPSN AUC for a community. This metric measures the false-positive rates of the model with respect to a community. A higher value means that a model is less likely to confuse between the normal post that mentions the community with a toxic post that does not.

- BNSP (Background Negative, Subgroup Positive) AUC: Here, we select toxic posts that mention the community and normal posts that do not mention the community, from the test set. The ROC-AUC score for this set will provide us with the BNSP AUC for a community. The metric measures the false-negative rates of the model with respect to a community. A higher value means that the model is less likely to confuse between a toxic post that mentions the community with a normal post without one.

For each of the three metrics above, we then employed a technique created by the Google Conversation AI team to combine the subgroup AUC calculations into one overall measure called the GMB (Generalized Mean of Bias) AUC which roughly reflects the average AUC metric across all minority subgroups. We report the following three metrics for each of our models

- GMB-Subgroup-AUC: GMB AUC with Subgroup AUC as the bias metric

- GMB-BPSN-AUC: GMB AUC with BPSN AUC as the bias metric

- GMB-BNSP-AUC: GMB AUC with BNSP AUC as the bias metric

EXPERIMENTS

Experimental Design

To start our augmentation experiments, we take the 19,229 example posts with agreed classification and split it into train, development, and test sets using a 80:10:10 ratio. We performed a stratified split across the three classes to maintain the balance of the original dataset. This split was retained for the baseline model and all augmentation experiments. This was done to ensure that all validation and test results were evaluated against the same examples in order to reduce the effect of the randomness in the split. This also ensured no information from the test and validation samples were leaked into the training phase.

A baseline BERT classification model was created, using the unaugmented training set for fine tuning.

Method	Text
Original Text - post_id: 1160689817224654848_twitter	after an update phoenix gang members can no longer attack the weaponsmaster
RD: Random Deletion	an update phoenix gang members attack
RS: Random Swap	longer an update attack weaponsmaster no phoenix members can the gang after
SR: Synonym Replacement (EDA, wordnet)	after an update capital of arizona crew extremity can no farsighted plan of attack the weaponsmaster
CS: Contextual Replacement (NLPAUG, BERT Base embeddings)	after both update phoenix gang group can no longer attack the house
RI: Random Insertion (EDA, wordnet)	after an update genus phoenix gang members can no longer attack the weaponsmaster
CI: Contextual Insertion (NLPAUG, BERT Base embeddings)	after an online update an phoenix gang members can no longer physically attack the weaponsmaster

TABLE III: Example Augmentations by Method

For each data augmentation method that we wanted to test, we augmented the training set by generating 2, 5, 7, and 10 additional examples. This allowed us to test both the method and the degree of augmentation.

Model Fine Tuning

We use the BERT base uncased pre-trained model from HuggingFace as a starting point for our classification. This is consistent with the base BERT model from the original HateXplain paper, allowing for a direct comparison of baselines. We set the maximum token length to 128, staying with the value used in the paper, this should be sufficient due to the nature of our dataset. Our training strategy was to allow for a large number of epochs and implement early stopping while monitoring validation accuracy. Early stopping is an effective technique to get the most out of our fine-tuning time (Dodge, et. al. 2020). Most of the model fine-tunings ran for between 9 and 12 epochs.

All training was done in a local environment with GPU acceleration. Fine tuning the baseline took around 15 minutes, with the augmented experiments taking between 30 and 50 minutes each. All fine tuning was done with the same hyperparameters for consistency. We were able to consistently reproduce similar baseline model performance as seen in the paper with the following hyperparameters:

Learning Rate: 2e-5 Batch Size: 64 Hidden Layer Dropout: 0.4 Early Stopping Patience: 4

Model Performance Results and Analysis

The main results we obtained are reported in Figures [1-3]. We observed that the augmentation techniques improved model performance across the accuracy, macro F1 and ROC AUC Scores. We also observed that increasing the size of the training set through augmentation did not always improve model performance. This is in line with our expectations, as data augmentation has been seen to improve model performance mainly on smaller data sets (Wei and Zou 2019). The EDA random swap operation performed well on all 3 metrics for various

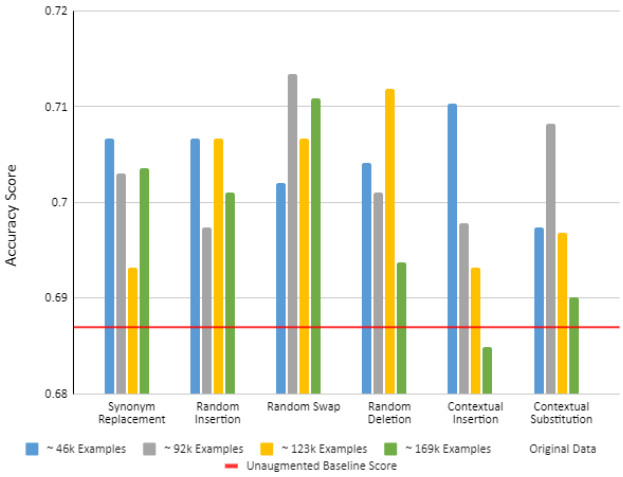

FIG. 1: *Accuracy Score*

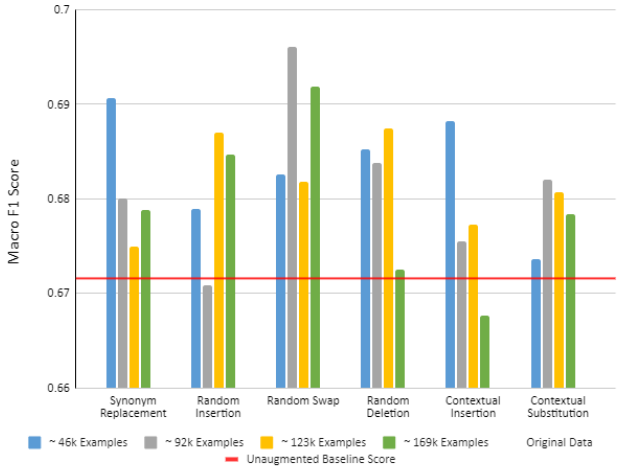

FIG. 2: *Macro F1 Score*

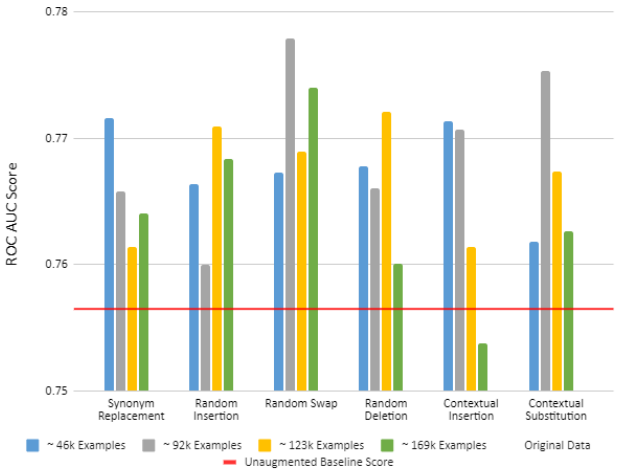

FIG. 3: *ROC AUC Score*

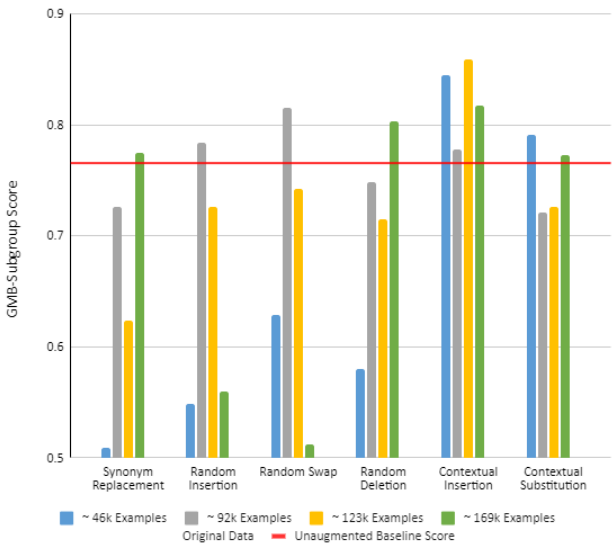

FIG. 4: *GMB-Subgroup Score*

sizes of training examples used. In the case of the NLP AUG Contextual Insertions we observed model performance to decrease as the size of the training set increased, even below baseline scores across the three metrics for about 169,000 training examples. We believe that these results indicate that for larger training set sizes, the model began to overfit on the training set and adding further training examples would have no improvement on the model's accuracy.

Model Bias Results and Analysis

The main bias scores are reported in Figures [4-6]

In general, we do not see strong evidence that non-contextual data augmentation techniques when applied to the training set for our BERT model resulted in improvements in the model's ability to give unbiased results, and in most cases we saw lower bias results from these techniques compared to the baseline. However, contextual augmentation techniques generally were seen to perform better than their non-contextual counterparts, and in some cases were able to beat the baseline score. This leads us to believe that randomly adding, removing, or replacing words in training set examples tends to increase the level of bias displayed by the model, while adding or substituting words that are similar to those around them tends to maintain or decrease the level of bias displayed by the model, as the model does not learn to take into account words that may be completely foreign to the examples it is learning from.

Similarly, we do not see a strong relationship between training set size and bias scores for any variant of training set augmentation. From this we conclude

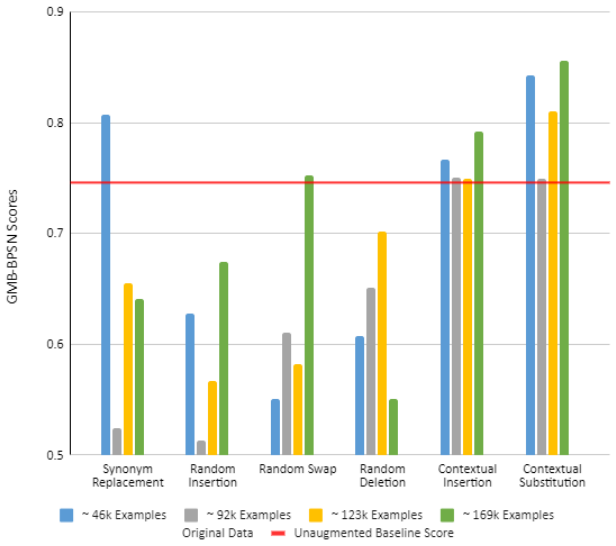

FIG. 5: *GMB-BPSN Scores*

that more training examples have little to no effect on the model's ability to produce unbiased results.

Error Analysis

We looked into the errors our models were making to understand how to improve the performance of the models and understand the limitations of this approach. The training sets have randomness in-

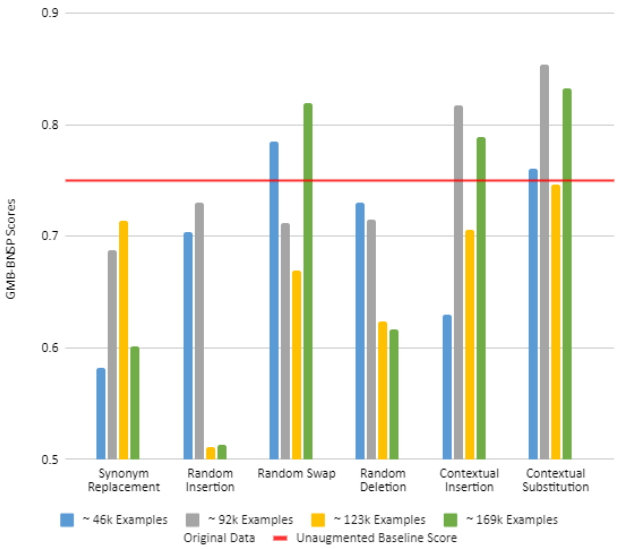

FIG. 6: *GMB-BNSP Scores*

duced with the data augmentation which is exacerbated with increasing the number of augmented examples used in training. The augmentation techniques themselves have probability weights that induce randomness in the results of the augmented sentences generated. These sources of randomness have a potential to cause errors in the models predictions.

We looked at examples of misclassified texts to try and understand why the models made those predictions:

"diversity is a code word for white genocide immigrants contribute and integrate 3 rd world retards freeload and fornicate" - predicted as offensive speech, correct label is hate speech. We can see in this example the absence of the common hate speech slurs in our dataset allowed the model to detect the offensive nature of this example but missed the hate content.

"¡user¿ terrorist protesters all are bangladeshi muslims" - predicted as normal speech, correct label is offensive speech. This example lacks context so the model misclassified it. It may also be affected by the inclusion of the ¡user¿ tag within the sentence.

"why are only white nations being invaded buy non white immigrants" - predicted as offensive speech, correct label is normal speech. This is a more interesting example as the annotation labels it as normal speech but it can be argued that this text is offensive in nature. There is the possibility of inaccuracies in human annotation especially with historical or cultural references specific to a niche group.

FUTURE WORK

With more time, we would hope to expand our project in several ways. As our primary experimental variable in this analysis was the method of training set augmentation, we would hope to employ other novel text augmentation techniques to test whether performance can be improved beyond what we have shown here. Similarly, we would be curious to test model performance using training sets augmented with the same techniques as we have given results for above albeit using different parameters (for example using various probabilities for whether a word is replaced, added or removed).

Additionally, we would hope if given more time to supplement our model analysis by calculating explainability metrics comparable to those presented by the authors of the original HateXplain paper using LIME or another similar technique. We also believe there to be some inaccuracies in the labeling of both the target group and the target label of some data points which may be on account of using "majority-rules" to decide on the final labels when choosing amongst the responses of the three annotators. Further work would benefit from a QA check of a sampling of data points to ensure accuracy in labeling. Lastly, we would have liked to re-run the training of our models several times using different randomized train-dev-test splits to establish an error band around which optimal model performance lies.

CONCLUSION

Using the HateXplain dataset, we constructed a model using a BERT transformer architecture which was able to classify examples of online text as either non-offensive, offensive, or hate speech. We were able to match the baseline performance shown in the original paper. We then performed a series of tests upon this model augmenting the size of the training set with examples which were generated by six textual augmentation techniques. Our results showed that all of the augmentation techniques we employed improved the accuracy of our model by 1-2% at lower multiples of the original training set size (2-5x) but that increasing the size of the training set beyond this led to mixed results with most models showing evidence of overfitting. We saw comparable results when evaluating model performance with macro F1 and ROC AUC Scores, which is not unexpected as we did not have a large class imbalance in our training set.

We were also able to evaluate the level of bias displayed by each of our experimental models and saw

that in general using non-contextual augmentation techniques increased the level of bias in our models' predictions while using contextual augmentation techniques improved or had no effect on the level of bias in our model's predictions. While we hypothesize that these results are due to the random nature of non-contextual augmentation techniques, further analysis is required before general conclusions can be drawn from these results.

Taken together, we can conclude that data augmentation techniques are effective at marginally improving the accuracy of hate speech classification models and can be a valuable tool in practice to detect and classify hateful online content.

BIBLIOGRAPHY

[1] Rui Cao and Roy Ka-Wei Lee. 2020. HateGAN: Adversarial Generative-Based Data Augmentaion for Hate Speech Detection. In the Proceedings of the 28th International Conference on Computational Linguistics, pages 6327-6338, Barcelona, Spai. International Committee on Computational Linguistics.

[2] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. "Automated hate speech detection and the problem of offensive language." In Proceedings of the International AAAI Conference on Web and Social Media, vol. 11, no. 1, pp. 512-515.

[3] Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial Bias in Hate Speech and Abusive Language Detection Datasets. In Proceedings of the Third Workshop on Abusive Language Online, pages 25–35, Florence, Italy. Association for Computational Linguistics.

[4] Jesse Dodge , Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. arXiv:2002.06305 [cs].

[5] Wyatt Dorris, Ruijia (Roger) Hu, Nishant Vishwamitra, Feng Luo, and Matthew Costello. 2020. Towards Automatic Detection and Explanation of Hate Speech and Offensive Language. In Proceedings of the Sixth International Workshop on Security and Privacy Analytics (IWSPA '20), March 18, 2020, New Orleans, LA, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3375708.3380312.

[6] Steven Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, Eduord Hovy. 2021. A Survey of Data Augmentation Approaches for NLP. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021.

[7] Anna Schmidt and Michael Weigand. 2017. A Survey on Hate Speech Detection using Natural Language Processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

[8] Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and William Campbell. 2019. A closer look at feature space data augmentation for few-shot intent classification. arXiv preprint arXiv:1910.04176.

[9] Edward Ma. 2019. NLP AUG. https://github.com/makcedward/nlpaug. (2022)

[10] William Yang Wang and Diyi Yang. 2015. That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using petpeeve Tweets. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2557–2563, Lisbon, Portugal. Association for Computational Linguistics.

[11] Mieradilijiang Maimaiti, Yang Liu, Huanbo Luan, Zegao Pan, and Maosong Sun. 2021. Improving Data Augmentation for Low-Resource NMT Guided by POS-Tagging and Paraphrase Embedding. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 20, 6, Article 107 (November 2021), 21 pages. DOI:https://doi.org/10.1145/3464427

[12] Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of Hate Speech in Online Social Media. In Proceedings of the 10th ACM Conference on Web Science (WebSci '19). Association for Computing Machinery, New York, NY, USA, 173–182. DOI:https://doi.org/10.1145/3292522.3326034.

[13] Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621.

[14] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. arXiv preprint arXiv:2012.10289v1.

[15] Vigna, F. D.; Cimino, A.; Dell'Orletta, F.; Petrocchi, M.; and Tesconi, M. 2017. Hate Me, Hate Me Not: Hate Speech Detection on Facebook. In Proceedings of the First Italian Conference on Cybersecurity, volume 1816, 86–95. Venice, Italy: CEUR-WS.org.

[16] 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, Hong Kong, China. Association for Computational Linguistics.

[17] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The Risk of Racial Bias in Hate Speech Detection. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.