Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 4523–4531
Marseille, 20-25 June 2022

Exploring Data Augmentation Strategies for Hate Speech Detection in Roman Urdu

Ubaid Azam, Hammad Rizwan, Asim Karim
Syed Babar Ali School of Science and Engineering, Lahore University of Management Sciences,
Lahore, Pakistan
{19030040, hammad.rizwan, akarim}@lums.edu.pk

Abstract

In an era where social media platform users are growing rapidly, there has been a marked increase in hateful content being generated; to combat this; automatic hate speech detection systems are a necessity. For this purpose, researchers have recently focused their efforts on developing datasets, however, the vast majority of them have been generated for the English language, with only a few available for low-resource languages such as Roman Urdu. Furthermore, what few are available have small number of samples that pertain to hateful classes and these lack variations in topics and content. Thus, deep learning models trained on such datasets perform poorly when deployed in the real world. To improve performance the option of collecting and annotating more data can be very costly and time consuming. Thus, data augmentation techniques need to be explored to exploit already available datasets to improve model generalizability. In this paper, we explore different data augmentation techniques for the improvement of hate speech detection in Roman Urdu. We evaluate these augmentation techniques on two datasets. We are able to improve performance in the primary metric of comparison (F1 and Macro F1) as well as in recall, which is impertinent for human-in-the-loop AI systems.

Keywords: Hate Speech Detection, Data Augmentation, Sentiment Classification

1. Introduction

In recent years, there has been an exponential increase in the number of users on social media platforms such as Twitter, Facebook, etc. Users love to share their opinions and emotions publicly on these platforms and interact with other users. These platforms provide users with the ability to disseminate information to millions of people worldwide within seconds. Users are granted freedom of speech to post whatever they desire given that it is within respectable bounds; but a lot of users also negatively exploit this freedom. Hateful content on these platforms is on the rise. Users are being targeted based on gender, religion, race e.t.c. They are being subjected to humiliation and bullying, which can cause rather adverse side effects such as psychological harm, which can result in increased amounts of anxiety experienced and in the worst case has led users to commit suicide. According to (Hinduja and Patchin, 2019) the chance of committing suicide increases two-folds if a person is subjected to cyberbullying. Social media platforms are now actively finding and removing hateful content by utilizing reporting mechanisms with content moderators and AI-based models. Although they have been able to clamp down on a significant portion of hateful content but the clampdowns influence can only be largely seen on hateful content generated in the English language and for regions where English is primarily spoken. A recent report by Aljazera[1] suggests that Facebook is one of the most widely used social media platforms has very few moderates for one of the most populous country (India) in the world and atop that, their AI-based models fail to detect hateful content in the regional language Hindi for which various dataset in this regard are publicly available for example (Kumar et al., 2018), (Feng et al., 2021) and (Bohra et al., 2018). For lower resource languages and countries that are not a priority, the results would be even more bleak.

Moreover, according to critical analysis performed by (Arango et al., 2019) currents AI models that are trained in a supervised training scheme for hate speech detection achieve commendable performance but only within a specific dataset. When they are utilized in real-world applications, these models fail miserably as they have been overfitted on datasets that lack diversity which is inherent of most of the publicly available datasets as they have been sampled over a short period i.e less than 6 months. Furthermore, these top-performing models from related literature are based on deep learning for example (Zimmerman et al., 2018), (Glazkova et al., 2021) and (Banerjee et al., 2021) which require a large amount of training data in order to achieve acceptable results. These issues present a challenge for hate speech detection in low resource languages such as Roman Urdu which is the focus of our work. Roman Urdu is the writing of Urdu language using Latin script, this is mostly used in social media platforms by users instead of its Peso Arabic script.

To solve the previously stated issues, the straightforward approach would be to gather more data and have it annotated but this can be time-consuming, costly and depending on the labelling scheme, acquiring annotations can become increasingly difficult. For developing nations the costs can be reason enough to just turn a

[1] https://www.aljazeera.com/news/2021/10/25/facebook-india-hate-speech-misinformation-muslims-social-media

blind eye to such content as money and resources can be better invested in other sectors such as infrastructure. Moreover, for datasets that are constructed in the English language researchers usually utilize Crowdsourcing platforms such as Amazon Mechanical Turk and Crowd-Flower but for regional languages, there are a limited number of workers on such platforms who specialize in the language and can understand cultural and regional contexts that are vital for correct annotations. Given these issues, there is a need for techniques to be explored which provide models with more training data to improve their generalizability by exploiting already available datasets.

In this paper, we explore data augmentation strategies for Roman Urdu. We explore several techniques from simple easy data augmentation (EDA) based augmentations to transformer based text generation and word replacement based augmentation for two Roman Urdu datasets using the same models and performance metrics that the datasets have been benchmarked on. We also gauge the models performance from the view of human-in-the-loop-AI based systems which have proven to the most successful systems deployed in the real world. For hate speech detection where a majority of content is not hateful, it would be best to have AI based models flag hateful content which would be then reviewed by human reviewers. This method would result in an increment in the human oversight as compared to complete automation but this would help reduce the number of errors made by the automated system. Most literature in hate speech detection explores complete automation therefore use some form of F1 measure to gauge their models performance but human-in-the-loop-AI systems require a focus on the models recall as flagging more not hateful text as hateful is less dangerous as compared to tagging hateful texts as not hateful.

2. Related Work

2.1. Hate Speech

In recent years, hate speech detection has gained the interest of researchers worldwide. A lot of research has been carried out in terms of creating models and custom datasets. The challenges in hate speech detection arise from the fact that hate speech content changes with demographics, thus custom datasets are designed to reflect the cultural and societal issues that are relevant for the particular demographic or country for which the work is being carried out. Early work by (Waseem and Hovy, 2016) involved classification between three labels racism, sexism and neutral, the authors used various features such as n-gram, text length etc to benchmark the dataset. As time went by researchers started experimenting with machine learning and deep learning based approaches along with ensembling approaches, (Badjatiya et al., 2017) and (Pitsilis et al., 2018) developed deep learning based ensemble approaches using which they were able to achieve high performance on the pre-mentioned dataset. As models started achieving human-like performance on hate speech datasets attention diverted to how hate speech is defined, (Davidson et al., 2017) discussed this problem, they argued that hate speech should be differentiated from offensive speech and showed that lexicon based approaches failed at this task as it assigns one label to all texts containing a particular hateful term. Similarly, (Agrawal and Awekar, 2018) presented a new dataset where they differentiated hate speech from cyberbullying and benchmarked the dataset using various deep learning based approaches. (MacAvaney et al., 2019) extended their work and discussed various approaches for automated hate speech detection, and they presented a multi-view SVM approach that produced great results in the identification of hate speech.

Recently, challenge tasks in hate speech detection have started becoming common; for example, offensive language detection at SemEval (Zampieri et al., 2019) and HASOC at FIRE (Mandl et al., 2021). These tracks provide datasets in one or more languages along with labels of various granularity in a set of three subtasks. Teams take part in this challenge to develop various models in a bid to achieve the highest performance. Apart from this, researchers are also exploring model explainability or interpretability (Mathew et al., 2021), (Pavlopoulos et al., 2021), these involve detecting text span that makes a sample text hateful and the importance of each word in a sample text for a specific prediction.

Although a majority of the previous work on hate speech detection has been focused on the English language, recently there has been a significant increase in the datasets available for low resource languages.

2.2. Roman Urdu Datasets

In this section, we discuss the Roman Urdu datasets, their evaluation metrics, and the models used.

(Rizwan et al., 2020) presented a lexicon for hateful words which they utilized to collect Twitter data to create their dataset. The annotated dataset is called the Roman Urdu Hate-Speech and Offensive Language Detection (RUSHOLD)[2] dataset which consists of 10,012 tweets. The authors have presented their dataset on two levels of annotations: namely ”fine-grained” and ”coarse-grained”. The ”fine grained” annotations consist of 5 different labels, the labels and their respective sample counts are the following: Abusive/Offensive (2,402), Sexism (839), Religious Hate (782), Profane (640), Normal(5349). For ”coarse-grained”, all labels except Normal are merged into one for a binary classification task. To benchmark the dataset the authors evaluated various multilingual embeddings and multilingual transformer models as baseline approaches along with state of the art models from related literature, especially those from Roman Hindi as it is very similar to RU. The authors have also presented a con-

[2] https://github.com/haroonshakeel/roman_urdu_hate_speech

Operation	Sentence
None	ishko seedha jannat bhejo allah ka order hai bahanchod [send him to paradise, it's the order of God sisterfu***r]
Synonym Replacement	**abeee** seedha **janat dekhiye** allah ka order hai **matherchod** [**now show** him the paradise, it's the order of God **motherfu***r**]
Random Insertion	ishko **abee** seedha **hei** jannat bhejo allah ka order hei **jiska** hai bahanchod **madharchood** [**send him directly to paradise right now** it's the order of God, he is sisterfu***r **motherfu***r**]
Random Swap	**bhejo** seedha jannat **order** allah ka **bahanchod ishko** hai [**directly send in paradise**, it's the order of God to him sisterfu***r]
Random Deletion	ishko seedha jannat bhejo allah ka hai bahanchod [send him to paradsise, **he is of God** sisterfu***r]
MT5 text generation	jannat bhejo allah ka order hai **bahanchodo allah ki aag se bachne ke liye ishki seedha jaanchod** [send to paradise it's the order of God, **sisterfu***rs to be safe from God's fire directly take his life**]
MBERT MLM	**ishko kaiii bhejo** allah ka order hai bhenchod [**send him somewhere**, it's order of God sisterfu***r]

Table 1: Data Augmentation Samples

volutional neural network (CNN) based deep learning model that builds atop various pretrained multilingual embeddings, they have named their model "CNN-gram". The model performs much better than other approaches but noted that simple transfer learning capabilities of transformer models showed considerable performance. The major evaluation metric of comparison is Macro F1.

RUT (Roman Urdu Toxic) is another RU hate speech dataset which consists binary labels i.e toxic and non-toxic is presented by (Saeed et al., 2021)*(the dataset was made available on requesting the authors)*. They have collected dataset samples from multiple social platforms like YouTube, Facebook, Twitter and have manually annotated them for a binary classification task. The dataset consists of 72,771 samples with 13,097 labelled as toxic samples and 59,674 as non-toxic. For modelling the dataset the authors perform in-depth experiments using various baselines such as support vector machine (SVM) along with various recurrent neural network and convolution neural network based models from related literature in a 5 fold validation scheme. The authors also experiment with ensemble based approaches using the best performing baseline models. Their ensemble consisting of BiLSTM, BiGRU and SVM is able to achieve top performance. The major evaluation metric of comparison is the F1 score.

2.3. Data Augmentation

Although the above stated two dataset are available for RU, the hate speech samples are low, for RUSHOLD although the hateful tweets are 46.57 percent of the dataset theses are only 4663 tweets and for RUT the hateful tweets represent only 18 percent of the dataset. This presents a significant hurdle in training deep learning. To improve performance of such models a lot of effort is being spent on various data augmentation techniques as access to more annotated data maybe not possible due to restraints in the real world e.g. financial costs. In computer vision, data augmentation is much easier as compared to NLP which requires that grammar to be preserved in order to keep the semantics of the sentence intact.

Data augmentation in NLP consists of a variety of techniques, from deleting words, adding punctuation, changing word positions e.t.c. (Wei and Zou, 2019) formalize these sets of techniques that are widely used nowadays in NLP for data augmentation and for boosting performance for text classification tasks. They have named this formalization as EDA; this consists of four techniques, i.e., synonym replacement, random insertion, random swap, and random deletion. In order to conserve the semantics of the texts and their label, a control parameter named α is used which controls the percentage of words in a text to be changed in an augmentation technique. Apart from simple techniques, techniques based on trees and parsing have also been used; (Şahin and Steedman, 2018) presented a data augmentation method based on dependency tree morphing. They proposed a crop approach in which they cropped sentences by removing dependency links. Secondly, they proposed a rotate approach in which they rotated the sentences by moving the tree fragments around the root. Their results show that their text augmentation technique works well on low resource languages, which are rich in case-marking systems. This however cannot work on most romanized languages as there are no grammatical rules and parts-of-speech can be difficult to extract.

Aside from simple augmentation techniques, deep learning based augmentation techniques have also been experimented with.(Marivate and Sefara, 2020) utilize various techniques such as synonym replacement using Word2Vec, word mix-up and round trip translation. They conclude that Word2Vec based augmentation is a viable option when one doesn't have access to a formal synonym model and that word mix-up is a viable technique for data augmentation as it reduces the effect of overfitting on deep learning models. They also concluded that round trip translation is hard to implement due to cost. (Kobayashi, 2018) proposed a data augmentation technique named contextual augmentation for labelled texts. In this approach, they experimented with using bidirectional contextual models for replacing words rather than using some form of synonym replacement that relies on dictionaries or word embeddings. (Park and Ahn, 2019) presented a similar idea, their augmentation technique is based on self-supervised learning. They have presented a label-masking language model (MLM) which utilizes BERT'S pretraining masked language model training task along with label specific tokens for masking while training. This label specific masking allows them to get words that are more likely to be used in one within that label while performing augmentations.

Recently, data augmentation has caught the interest of hate speech detection researchers. (Wullach et al., 2020) presented a dataset of one million realistic hate and non-hate sequences produced by the generative language model via GPT-2. They used this dataset to train a deep learning-based hate speech detector which improved the performance of other hate speech datasets. Although, the generator model is trained for English only, the method can be generalized to other language given that such a large generative model is available for said language. Similar work has been done by (Cao and Lee, 2020), they propose a generative adversarial network (GAN) based model named HateGAN. A generator and a discriminator are trained iteratively for hateful text generation. The generated texts are diverse, coherent, and relevant to hate speech detection. Moreover, their results indicated that generative models allowed them to tackle the challenge of imbalance classes which is inherent of hate speech datasets.

3. Data augmentation Strategies

The augmentation techniques and their respective implementation details are stated below:

3.1. Synonym Replacement and Random Insertion

Both synonym replacement and random insertion have one step in common, which is that they require a means to find similar words. For this purpose, a variety of techniques can be used such as a language dictionary for example word net (Miller et al., 1990) and embedding based approaches such as word co-occurrence based embeddings (e.g. word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014)) and deep learning based embeddings such as ELMo and Transformers (e.g. BERT (Devlin et al., 2018)) to find similar words in vector space.

RU lacks proper language dictionaries and training deep learning based embeddings requires significant computational resources to train properly and open-source multilingual model embeddings don't work well for our use case. Hence we have opted to finetune the Word2Vec embeddings made available by RUSHOLD dataset authors. Their embeddings are trained on more than 4.7 million tweets collected from Twitter, we further finetune these embeddings on an open-source dataset made available at [3]. The dataset is not published and is not properly benchmarked hence we do not use it in our evaluations.

To perform synonym replacement, we initially choose words randomly from the input sample to be replaced. For each word we find the top 5 similar words using Word2Vec embeddings with cosine similarity as a scoring metric. From these 5 words, we randomly choose 1 for replacement. For random insertion, we perform the same steps except rather than replacing the words in the sample we insert the new words somewhere randomly in the sample. The choice to choose one of the 5 similar words allows us to bring more versatility to the dataset. To choose the number of words that are replaced or inserted is controlled as a hyperparameter.

3.2. Random Swapping

Random swapping involves randomly choosing two words in the input sample and swapping their respective positions. To control the number of swap we use a hyperparameter as well.

3.3. Random Deletion

Random Deletion involves randomly deleting words in an input sample. Removing hateful words in a sample can lead to a change in its target/label. Thus to avoid this from occurring we use a lexicon based approach to stop them from being deleted. For the RUSHOLD dataset, we utilize the lexicon that is made available by the authors, for RUT we create this lexicon by manually analysing the vocabulary for their dataset.

For each input sample, we assign the words a score between 0 - 1. We remove all words that lie below a certain threshold, the value of the threshold is controlled as a hyperparameter. This technique allows for the removal of hateful words as well but we use a hard block in case there is only one hateful word left in the input sample.

3.4. MT5 Text Generation

MT5 is a large multilingual text-to-text transformer that has been trained on a variety of tasks such as paraphrasing, sentence completion. We experiment with creating new hateful samples via conditional generation where an input sample from the dataset is conditioned on.

[3] https://github.com/shaheerakr/roman-urdu-abusive-comment-detector

Model	Accuracy	Precision	Recall	Macro F1-score
XLM-Roberta (rizwan et al.)	79.0	70.0	**75.0**	72.0
XLM-Roberta (Synonym Replacement)	**82.0**	75.0	72.0	73.0
XLM-Roberta (Random Swap)	81.0	**75.0**	74.0	**74.0**
XLM-Roberta (Random Insertion)	80.0	75.0	69.0	72.0
XLM-Roberta (Random Deletion)	81.0	75.0	72.0	73.0
XLM-Roberta (MT5 Text Generation)	81.0	73.0	72.0	73.0
XLM-Roberta (MBERT MLM)	80.0	72.0	69.0	70.0
Multilingual BERT (rizwan et al.)	77.0	72.0	65.0	67.0
Multilingual BERT (Synonym Replacement)	**81.0**	**73.0**	**73.0**	**73.0**
Multilingual BERT (Random Swap)	79.0	72.0	69.0	71.0
Multilingual BERT (Random Insertion)	79.0	73.0	71.0	72.0
Multilingual BERT (Random Deletion)	76.0	67.0	70.0	68.0
Multilingual BERT (MT5 Text Generation)	76.0	71.0	70.0	69.0
Multilingual BERT (MBERT MLM)	77.0	68.0	69.0	68.0
BERT+CNN-gram (rizwan et al.)	82.0	75.0	74.0	75.0
BERT+CNN-gram (Synonym Replacement)	82.0	74.0	73.0	74.0
BERT+CNN-gram (Random Swap)	83.0	75.0	73.0	74.0
BERT+CNN-gram (Random Insertion)	82.0	75.0	72.0	73.0
BERT+CNN-gram (Random Deletion)	83.0	76.0	73.0	74.0
BERT+CNN-gram (MT5 Text Generation)	**83.0**	**77.0**	**75.0**	**76.0**
BERT+CNN-gram (MBERT MLM)	83.0	76.0	74.0	75.0

Table 2: Results of data augmentation strategies on the RUSHOLD dataset

3.5. Word Replacement via Multilingual BERT Masked Language Modelling (MBERT MLM)

For this approach, we randomly mask some of the words of an input sample and use multilingual BERT to predict the masked words. Similar to our approach in synonym replacement and random insertion instead of choosing the word with the highest probability we choose randomly between the top 5 words with the highest probability. Although this allowed for more variations than just choosing the top word, the quality of the produced samples are of inferior quality.

To mitigate this issue we use the same dataset that we used to train our Word2Vec model and randomly sampled Twitter tweets to finetune BERT in its default masked language modelling task. This results in samples that have more word variations and are of higher quality.

4. Experimental Setup

In this section, we describe the details of our data augmentation experiments for both RUSHOLD and RUT datasets.

For both datasets, we performed an initial hyperparameter search for our augmentation strategies using baseline models from both datasets. For synonym replacement, random insertion and word replacement via MBERT MLM, we explore the ratio of words to apply the operation. We tested percentages between 0 and 1 with an increment of 0.1 and found that the percentage of 0.2 gives us the best results; Increasing the percentage reduces performance and lowering the percentage has minimal effect on the results. Similarly, for random deletion, we span the threshold between 0 and 1 and find that 0.4 works best. For random swapping we experiment between 1 and 5 total swaps, the results show that 2 swaps are optimal. For text generation via MT5 , we set the minimum text length to be 50 characters and a maximum length of 100 characters, which is about 10 to 30 words.

To compare the results of data augmentation strategies, we chose the best performing model and the baselines for both datasets. For the RUSHOLD dataset the baselines are multilingual BERT and XLM-RoBERTa trained for sentiment classification and for RUT the baselines are SVM, LR, BiLSTM and BiGRU. SVM and LR both use TF-IDF as the input feature while both BiLSTM and BiGRU use the fastText skipgram embeddings. The top performing model for RUSHOLD is CNN-gram with multilingual BERT embeddings and RUT an ensemble of SVM, BiLSTM and BiGRU.

For RUSHOLD the authors have made available the train, test and validation splits. The number of samples in split is 7209, 2003, 801 respectively. For RUT, the authors have provided seed that they have used for 5 fold cross validation and the seed for breaking the train fold into train and validation sets. The train,test and validation sets have 49483, 14555, 8733 samples respectively. For both datasets we perform augmentations on the train folds and report the results on the same evaluation metrics that they the authors have used. For RUT, we report average performance on 5 folds with F1 score as the primary metric and for RUSHOLD we report the results on the preavailable

Models	Accuracy	Precision	Recall	F1-score
SVM (saeed et al.)	**93.87 ± 0.21**	82.69 ± 2.00	**83.56 ± 1.70**	**83.08 ± 0.30**
SVM (Synonym Replacement)	93.27 ± 0.38	83.61 ± 2.07	77.95 ± 0.62	80.66 ± 0.88
SVM (Random Swap)	93.27 ± 0.68	**92.20 ± 4.71**	68.89 ± 8.20	78.43 ± 3.53
SVM (Random Insertion)	93.72 ± 0.31	87.93 ± 1.65	75.54 ± 0.69	81.25 ± 0.82
SVM (Random Deletion)	92.67 ± 0.20	80.75 ± 1.53	77.91 ± 1.72	79.28 ± 0.56
SVM (MT5 Text Generation)	93.24 ± 0.22	82.58 ± 1.27	79.23 ± 1.59	80.85 ± 0.67
SVM (MBERT MLM)	93.15 ± 0.22	86.76 ± 2.07	73.20 ± 2.81	79.35 ± 1.03
LR (saeed et al.)	93.67 ± 0.18	83.98 ± 1.75	80.19 ± 1.91	82.00 ± 0.52
LR (Synonym Replacement)	93.19 ± 0.24	81.23 ± 1.49	80.97 ± 1.94	81.07 ± 0.71
LR (Random Swap)	**93.91 ± 0.18**	87.40 ± 1.46	77.36 ± 1.05	**82.06 ± 0.46**
LR (Random Insertion)	93.20 ± 0.41	80.79 ± 2.46	81.80 ± 2.13	81.25 ± 0.97
LR (Random Deletion)	93.00 ± 0.29	79.74 ± 1.22	**81.98 ± 1.43**	80.83 ± 0.79
LR (MT5 Text Generation)	93.37 ± 0.16	82.44 ± 1.26	80.36 ± 1.42	81.37 ± 0.42
LR (MBERT MLM)	93.51 ± 0.19	**85.63 ± 1.81**	76.93 ± 1.93	81.02 ± 0.59
BiLSTM (saeed et al.)	**95.05 ± 0.21**	89.78 ± 0.35	81.80 ± 1.20	**85.60 ± 0.70**
BiLSTM (Synonym Replacement)	92.69 ± 4.59	86.33 ± 3.63	84.13 ± 3.93	85.09 ± 0.80
BiLSTM (Random Swap)	94.76 ± 0.09	85.98 ± 2.56	84.92 ± 3.35	85.37 ± 0.40
BiLSTM (Random Insertion)	92.90 ± 4.61	87.71 ± 3.52	83.74 ± 4.22	85.51 ± 0.96
BiLSTM (Random Deletion)	94.56 ± 0.30	84.65 ± 1.22	**85.25 ± 1.24**	84.94 ± 0.81
BiLSTM (MT5 Text Generation)	94.79 ± 0.37	88.88 ± 2.80	81.36 ± 2.03	84.91 ± 0.90
BiLSTM (MBERT MLM)	94.81 ± 0.17	**90.52 ± 0.80**	79.56 ± 1.90	84.67 ± 0.74
BiGRU (saeed et al.)	**95.03 ± 0.25**	**90.77 ± 1.22**	80.59 ± 0.82	85.37 ± 0.72
BiGRU (Synonym Replacement)	92.74 ± 4.43	87.16 ± 1.82	83.08 ± 1.90	85.05 ± 0.79
BiGRU (Random Swap)	94.83 ± 0.23	86.33 ± 0.30	84.72 ± 1.66	**85.51 ± 0.79**
BiGRU (Random Insertion)	92.51 ± 4.45	88.57 ± 0.68	82.33 ± 1.25	85.33 ± 0.68
BiGRU (Random Deletion)	94.35 ± 0.21	83.94 ± 1.69	**84.95 ± 1.60**	84.41 ± 0.48
BiGRU (MT5 Text Generation)	94.92 ± 0.24	89.22 ± 2.14	81.77 ± 0.97	85.31 ± 0.49
BiGRU (MBERT MLM)	94.70 ± 0.24	88.95 ± 1.75	80.64 ± 1.46	84.57 ± 0.64
ML+B. Deep (MV) (saeed et al.)	**95.30 ± 0.20**	**90.41 ± 0.32**	82.64 ± 1.13	86.35 ± 0.66
ML+B. Deep (MV)(Augmented)	95.14 ± 0.34	86.76 ± 2.17	**86.28 ± 2.02**	**86.48 ± 0.88**

Table 3: Comparisons of baseline models with our augmented approaches on RUT dataset

data splits with Macro F1 as the primary metric.

5. Results and Discussion

In this section, we discuss the results of our augmentation techniques. The results of augmentation strategies is shown in Table 2 for RUSHOLD and Table 3 for RUT.

On the RUSHOLD dataset, the macro f1 score achieved by XLM-RoBERTa without data augmentation is 72.0, most augmentation techniques are able to improve upon this with an increment of one point. Random swapping is able to achieve the best result for all augmentation techniques with an improvement of two points. Moreover, the results are not all positive, for MBERT MLM augmentation the F1 score instead of increasing decreases to a value of 70.0. Although overall there is an increase in the Macro F1 scores, its rise is owed to an increase in precision rather than recall. The recall score for the baseline is 75.0, the recall for all augmentation techniques is lower than the baseline, which means that in this case more hate speech content would be missed if models trained on augmentation datasets were to be deployed.

For multilingual BERT synonym replacement is able to achieve a large increase of 6 points in the Macro F1 score over the non-augmented model and which is closely followed by random insertion with an increase of 4 points. All augmentation techniques are able to improve the model's performance over the baseline macro f1 score of 67.0. Moreover, the recall has also improved for all augmentation strategies. Similar to the trend in Macro F1, the highest recall is achieved by synonym replacement with a score value of 73.0 followed by random insertion with a score of 71.0. Although MBERT MLM augmentation is able to achieve improvement over the non-augmented model its performance is still the poorest yet again.

For BERT+CNN-gram the results are somewhat opposite to the ones observed for transformer models. For most augmentation techniques the Macro f1 falls by one or remains the same in the case of MBERT MLM. Although the macro f1 score for MBERT MLM is the same as the model trained without data augmentation, the model is able to improve its precision and accuracy. The top performing model, in this case, is the model trained on data augmented by MT5 text genera-

BiLSTM (saeed et al.)				BiLSTM (Random Insertion)		
	Non Toxic	Toxic			Non Toxic	Toxic
Non Toxic	**0.98**	0.02		Non Toxic	**0.976**	0.024
Toxic	0.184	**0.816**		Toxic	0.174	**0.826**

Table 4: Heat Map for BiLSTM

CNN-gram (rizwan et al.)					
	Abusive/offensive	Normal	Religious	Sexist	Profane
Abusive/offensive	**0.71**	0.11	0.04	0.07	0.07
Normal	0.04	**0.92**	0.02	0.01	0.01
Religious	0.09	0.11	**0.75**	0.04	0.01
Sexism	0.17	0.09	0.01	**0.72**	0.01
Profane	0.20	0.08	0.02	0.07	**0.63**

CNN-gram (MT5 Text Generation)					
	Abusive/offensive	Normal	Religious	Sexist	Profane
Abusive/offensive	**0.73**	0.13	0.04	0.05	0.05
Normal	0.04	**0.93**	0.01	0.01	0.01
Religious	0.10	0.11	**0.75**	0.03	0.01
Sexism	0.17	0.10	0.02	**0.70**	0.01
Profane	0.20	0.08	0.02	0.06	**0.64**

Table 5: Heat Map for BERT+CNN-gram

tion with a Macro F1 score of 0.76 an improvement of 1 point. For recall, the story is similar the top performing model achieves an increment of 1 point over the no augmented model, for MBERT MLM the results do not change and for the rest of the augmentation techniques performance has deteriorated.

On the RUT dataset, we observe mostly a mixed bag of results. The F1 score for LR at best has increased by a minuscule 0.06 points but for SVM there is a significant decrease in F1 score, as the maximum drop is 4.85 points on random deletion. Moreover, there are large variations in performance for machine learning models for different data augmentation techniques. Similarly for deep learning based models, BiGRU achieves a small increase performance which results in an increase of 0.14 points using random swapping augmentation but for BiLSTM there is a drop in performance which in worst case is a maximum drop of 0.96 points using random deletion. For the ensemble model we use the combination of the following three model: Svm (Random Insertion), BiLSTM (Random Insertion) and BiGRU (Random Swapping) as they achieved top performance with some augmentation techniques. We are also able to achieve better performance on the ensemble model as the F1 score has increased by 0.13. Comparing the machine learning models, the performance of deep learning models is overall much more stable for different data augmentation strategies.

From a deeper dive into the metrics it can be observed that the high F1 achieved by most models trained without augmenting the dataset is from high precision rather than high recall and that the increase in recall results in a reduction of the models precision. Which means that although more content is flagged as toxic, more toxic content is being classified correctly. For LR there is a 1.97 point increase in recall using random deletion over the baseline score of 80.19. For deep learning models, the largest increase is of 3.64 points on the ensemble model followed by BiLSTM which has an increase of 3.45 using random deletion and for GRU an increase of 4.36 points on random deletion as well.

Although for both dataset RUT and RUSHOLD, we have observed a mixed bag of results, nonetheless we are able to achieve a higher recall and F1 scores for both datasets which means that these techniques are viable to improve hate speech detection models. This can be seen more clearly from the heat maps for top performing models shown in figure 4 and 5 for RUT and RUSHOLD respectively. *(For RUT we have opted to use BiLSTM as creating the heat map for ensemble requires significant computation time to reproduce original results).*

6. Conclusion and Future Work

In this paper, we have explore different data augmentation techniques for the improvement of hate speech classification in Roman Urdu. We have evaluated these augmentation techniques on two Roman Urdu datasets. We are able improve performance in the primary metric of comparison as well as in recall which is impertinent for human-in-the-loop AI systems. In the future, we aim develop and evaluate more approaches not only for Roman Urdu but for other languages as well.

7. Bibliographical References

Agrawal, S. and Awekar, A. (2018). Deep learning for detecting cyberbullying across multiple social media platforms. In *European conference on information retrieval*, pages 141–153. Springer.

Arango, A., Pérez, J., and Poblete, B. (2019). Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 45–54.

Badjatiya, P., Gupta, S., Gupta, M., and Varma, V. (2017). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion*, pages 759–760.

Banerjee, S., Sarkar, M., Agrawal, N., Saha, P., and Das, M. (2021). Exploring transformer based models to identify hate speech and offensive content in english and indo-aryan languages. *arXiv preprint arXiv:2111.13974*.

Bohra, A., Vijay, D., Singh, V., Akhtar, S. S., and Shrivastava, M. (2018). A dataset of hindi-english code-mixed social media text for hate speech detection. In *Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media*, pages 36–41.

Cao, R. and Lee, R. K.-W. (2020). Hategan: Adversarial generative-based data augmentation for hate speech detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6327–6338.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. (2021). A survey of data augmentation approaches for nlp. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988.

Glazkova, A., Kadantsev, M., and Glazkov, M. (2021). Fine-tuning of pre-trained transformers for hate, offensive, and profane content detection in english and marathi. *arXiv preprint arXiv:2110.12687*.

Hinduja, S. and Patchin, J. W. (2019). Connecting adolescent suicide to the severity of bullying and cyberbullying. *Journal of school violence*, 18(3):333–346.

Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457.

Kumar, R., Reganti, A. N., Bhatia, A., and Maheshwari, T. (2018). Aggression-annotated corpus of hindi-english code-mixed data. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N., and Frieder, O. (2019). Hate speech detection: Challenges and solutions. *PloS one*, 14(8):e0221152.

Mandl, T., Modha, S., Shahi, G. K., Madhu, H., Satapara, S., Majumder, P., Schaefer, J., Ranasinghe, T., Zampieri, M., Nandini, D., et al. (2021). Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages. *arXiv preprint arXiv:2112.09301*.

Marivate, V. and Sefara, T. (2020). Improving short text classification through global augmentation methods. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 385–399. Springer.

Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., and Mukherjee, A. (2021). Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867–14875.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Park, D. and Ahn, C. W. (2019). Self-supervised contextual data augmentation for natural language processing. *Symmetry*, 11(11):1393.

Pavlopoulos, J., Sorensen, J., Laugier, L., and Androutsopoulos, I. (2021). Semeval-2021 task 5: Toxic spans detection. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 59–69.

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

Pitsilis, G. K., Ramampiaro, H., and Langseth, H. (2018). Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence*, 48(12):4730–4742.

Rizwan, H., Shakeel, M. H., and Karim, A. (2020). Hate-speech and offensive language detection in roman urdu. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2512–2522.

Saeed, H. H., Ashraf, M. H., Kamiran, F., Karim, A., and Calders, T. (2021). Roman urdu toxic comment classification. *Language Resources and Evaluation*, pages 1–26.

Şahin, G. G. and Steedman, M. (2018). Data augmentation via dependency tree morphing for low-resource languages. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009.

Waseem, Z. and Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Wei, J. and Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *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.

Wullach, T., Adler, A., and Minkov, E. (2020). Towards hate speech detection at large via deep generative modeling. *IEEE Internet Computing*, 25(2):48–57.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

Zimmerman, S., Kruschwitz, U., and Fox, C. (2018). Improving hate speech detection with deep learning ensembles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.