EDA for hate speech classification study

Shofolawe-Bakare Babatunde
181075578
Dr Matthew Purver
School of Electronics Engineering and Computer Science

***Abstract*—The ability of automated classification algorithms to successfully detect instances of hate speech, is dependent on their capacity remain robust when handling datasets in which hate speech is strongly underrepresented. This accounts for most available datasets, as hate speech represents a very small fragment of online information. Data imbalance poses a challenge to hatespeech classification hindering the effective performance of algorithms. This paper explores the use of easy data augmentation(EDA) techniques, to improve classifier performance on a highly imbalanced dataset at the expense of menial compute.**

***Index Terms*—Data augmentation, Data imbalance, Neural networks**

I. Introduction

A. Background

The massive growth of social media in the last two decades has brought with it a paradigm shift in the way we communicate and share information, considering its many positives it equally possesses obvious negatives. Offensive language and in particular hate speech is a problem of today's social media platforms with dire consequences such as poor psychological wellbeing, hate crime, and minority group prejudice in both virtual domains and local communities (for Global Community, 2019). Hate speech is a complex topic without a general accepted description, according to Davidson et al. (2017) its described as the use of language to express hatred and its associated sentiments to a specific group with the intention of humiliating or insulting its members. This is taken a step further by Sharma et al. (2018) whose definition highlights the speech's nature to incite violent action.

As the definition differs, from one school of thought to the other so also does approach used in various geographical regions to tackle it, In countries like the United Kingdom and Germany criminal prohibitions are imposed on hate speech (Theil, 2019), with offences like online racial abuse attracting legal penalties like fines and in some cases jail time. The United States on the other hand faces a free speech vs hate speech debacle as the right to free speech under the first amendment (van Mill, 2018), inhibits hate speech regulation. In the past five years there has a glaring increase hate related speech in the U.S. after Donald Trump's election (Alexis, 2016), likewise in U.K., Muslim minorities face more hate crime incidents since the Manchester and London Bridge attacks (Alan, 2017).

The need for crucial action has been recognised by governments, law enforcement and social media companies. Platforms like Facebook, Twitter, YouTube have started taking the necessary steps to monitor their platforms through the use of manual moderation and automated detection with algorithms(Simonite, 2020), (Natasha, 2015). In addition countries like Germany (Lomas, 2017) and the U.K. (Nicky, 2020) are creating regulations to hold social media platforms accountable.

B. Subject focus/Scope

Although active monitoring and policing of platforms manually has been partially replaced by automated systems, these models struggle with nuanced detection of hate speech in online content. This is because instances of hate speech occur less frequently when compared to offensive language as a whole. As a result datasets for hate speech are typically highly imbalanced, with hate speech making a small percentage of entire dataset. A scenario like this leads to overfitting and missclassification by the model in an instance were the detection of the smaller class is crucial, hence the need for more data for the less represented class.

Data augmentation is a popular means used in image classification problems to increase the amount of data available via various operations. While data augmentation isn't as popular in the Natural language processing(NLP) domain there have been a few notable instances of its use in recent years (Zhang et al., 2015), (Wang and Yang, 2015), (Shleifer, 2019).

C. Aims & Objectives

This paper proposes the use of the easy data augmentation (EDA) to enhance the performance of a classifier on a hate speech dataset. A simple LSTM Model with pretrained word embeddings is experimented with using augmented dataset variations.

D. Structure of paper

The subsequent sections following would examine relevant literature and work relevant to this paper(section2); describe the approach and methodology used, highlighting the chosen architecture, dataset, classifier pipeline and how it works (section 3); illustrate experiments performed with classification model and different data augmentation techniques comparing performance (section 4); discuss results, accentuating the problems encountered (Section 5); and conclude with unsolved problems worth tackling/future works (Section 6)

II. LITERATURE REVIEW

A. Overview

While great strides have been made in automated Hate speech detection, earlier traditional techniques treated the task as a binary classification problem (hate vs non hate). Burnap and Williams (2015) used a combination of three classifiers (probabilistic, rule-based, and spatial-based) with a voting ensemble meta classifier, to classify hateful or antagonistic speech, Kwok and Wang (2013) took a supervised learning approach to classify racist and non racist speech, they found a tweet was classified as racist in all examples 86% of the time if it contained offensive language. In addition they also observed subtle linguistic differences in grammar, for instance the word *(n*gga vs n*gger)* influenced classification, with the latter more likely to be classified as hate speech. Both authors Burnap and Williams (2015) and Kwok and Wang (2013) had a relatively high recall using a bag-of-words approach though this method made their models highly susceptible to misclassification of offensive tweets as hate speech.

Davidson et al. (2017) introduced a new approach handling the task as a multi-class problem, this improved performance(precision, recall, classification metrics) as well as accuracy of their model. They experimented with statistical methods like Logistic Regression(LR), linear Support Vector Machines(SVM), naive Bayes, decision trees and random forests. Whilst this was significant development, their annotated dataset which consisted of 24,783 labelled tweets grouped into (Hate, Offensive, Neither) had a high data imbalance as the hate class accounted for less than 6% of the labelled data. This in turn although providing high overall accuracy led to low recall for the hate class when compared to its other class counterparts (offensive and neither). Following (Davidson et al., 2017) which equally used statistical methods, Badjatiya et al. (2017) kicked off the use of deep learning techniques experimenting with (Convolutional Neural Network(CNN), Long Short-Term Memory(LSTM) etcetera), exploring its use for multi class hate-speech detection(racist, sexist or neither) and demonstrating its effectiveness.

B. Data Augmentation in Learning

Data imbalance within classes of a dataset is a natural problem present in real-world scenarios varying from fraud detection (Van Vlasselaer et al., 2015), to cancer prediction in medicine (Krawczyk et al., 2016). Strongly imbalanced datasets add a degree of difficulty which inhibits good model performance on machine learning problems. This is because abstractions learnt by the classification algorithm are skewed towards the majority class, so performance is poorer in the underrepresented class.

While other techniques like undersampling and oversampling mitigate this to some extent. Data augmentation is a particularly useful method to counter data imbalance and improve model performance. This is done by using available data to create additional differing data samples, with more data available as input for the model it is able generalise more and make better predictions when encountering new data test samples. This methodology is commonly used in the domain of computer vision, where images for example undergo rotational and geometrical transformations to create new image samples. It's also used in audio speech recognition, here speed perturbation applied to audio snippets creates new samples. In both vision (Krizhevsky et al., 2012), (Szegedy et al., 2015) and speech domains (Jaitly and Hinton, 2013), (Ko et al., 2015), data augmentation has shown noteworthy results.

C. Critical evaluation of works

NLP as a domain doesn't have a broad use of augmentation when compared other domains (vision & speech). This is likely because of the abundance of textual information on the internet with sources like Google and Wikipedia but issues with this concept arise when dealing with textual data which occur sparsely online. In addition augmenting text could pose more of a challenge, as semantic meaning must be retained when creating synthetic samples so as to avoid changing a text samples class which would then require manual relabelling. In recent years a some text based augmentation methods with the following themes shown below have proved to be significant:

1) Translation based Augmentation: Translation is a means by which a word, sentence or text is changed to another language. Translation of a text to another language is known as forward translation, and the reverse of initially translated text is known as back translation. Sennrich et al. (2015) used back translation to generate new data by translating source data into a new language and back into its original. Fadaee et al. (2017) employed a method called translational augmentation for a neural translation task with insufficient samples to create both new source(initial language) and target(new language) data. Both papers show an improved BLEU machine translation score, although a drawback for Sennrich et al. (2015) was that sentences incorrectly translated could have wrong labels and therefore reduce model performance.

2) Linguistic feature augmentation: As words make up text, textual data augmentation typically involves the removal, addition, or swapping of words, sentences or text to create new samples. This operations are usually done through the use of a dictionary(lexical database like word net). Mueller and Thyagarajan (2016) replaced words in sentences randomly with synonyms generating data for their Siamese RNN(Recurrent Neural Network). (Kolomiyets et al., 2011) on the other hand replaced head words with respective synonyms. Zhang et al. (2015) found suitable synonyms for their text understanding temporal convolutional network through the use of a geometric function.

3) Embedding based Augmentation: Similarly to linguistic augmentation, models within this space ,remove, add, or swap words, sentences or text to create new data. But instead do so through the use of word embeddings, representations of text which possesses semantic meaning. Wang and Yang

(2015) used k-nearest neighbour on word embeddings of sentences to find synonyms and replace actual words with similar word vectors for a multiclassification task it achieved an enhancement of a 1.4% in its F1-score.

4) Predicitve and Natural Language Generation(NLG) Augmentation: (NLG) is a process by which a system creates readable text(natural language) from structured data. (Hemker, 2018) employed the underlying idea here, to generate new synthetic data for hate speech classification, which possessed similar semantic patterns to original data. They trained a Recurrent Neural Network (RNN) model as a text generator to predict the next word in a sequence, based on a learnt probability distribution. This technique yielded minuscule gains in F1 score but required training a whole model which is computationally intensive to implement. On the other hand Kobayashi (2018) used a bi-directional language model to replace predicted contextual words paradigmatically.

Of all the works discussed (Hemker, 2018) is the most closely related to this paper as it attempts to improve a hate speech classification performance using state of the art deep learning topology as well as natural language generation, (threshold based and POS-tag) synonym replacement for data augmentation. While direct comparison cant be made with this work, it's worth noting they gained 2-3% improvement in overall accuracy and about 5% improvement in hate accuracy specifically with their best augmentation technique(threshold based synonym replacement)). They also mentioned it was relatively slow in implementation and hence quite time intensive. EDA on the other hand is simple to implement with interesting gains in hate-class and overall performance exceeding 5%.

III. METHODOLOGY

The following sub-sections describe the chosen dataset in greater detail, expands on the mechanics of the neural network architecture used, explains the data augmentation techniques employed, discusses the embedding method utilised and provides further information on the experimental setup approach.

A. Chosen Dataset

The dataset chosen for this task is the hatebase dataset Davidson et al. (2017) this dataset was annotated by crowd flower from a random sample 25,000 tweets. It is comprised of three labelled classes:

- Hate(0)
- Offensive(1)
- Neither (2)

Hate accounts for (5.77%) of the data set, Offensive(77.43%) and Neither(16.7%)

B. Neural Network Model architecture

An LSTM, which is a type of RNN is used. This is because of ability to capture long range dependencies via the use of its cell states and gates. In simpler terms it has memory to carry information over a long sequence, this is particularly useful when dealing with lengthy sentences.

It possesses a forget, input and output gate, coupled with cell state which stores information and a sigmoid function. The forget gate determines what input is forgotten or retained based on the sigmoid function, input gate help to update the cell state. The cell state is gets its value multiplied by the forget gate vector which decides whether its keeps its current value or drops it, after further down the line it takes output from the input gate via a point wise addition which updates it state. The output gate determines what the next hidden state would be

TABLE I
AUGMENTED DATA CORPUS

Data Set Corpus	Hate Corpus	Offensive Corpus	Neither Corpus
Original data - (24783)	**1430**	**19190**	**4163**
After Validation split - (4213)	258	3726	679
After Test split - (3718)	199	2894	625
After Train split - (16852)	973	13020	2859
Train + SR - (33704)	1946	26040	5718
Train + RS - (33704)	1946	26040	5718
Train + RD - (33704)	1946	26040	5718
Train + RI - (33704)	1946	26040	5718
Train + SR + RS - (50556)	2919	39060	8577
Train + SR + RD - (50556)	2919	39060	8577
Train + SR + RI - (50556)	2919	39060	8577
Train + RS + RD - (50556)	2919	39060	8577
Train + RS + RI - (50556)	2919	39060	8577
Train + RD + RI - (50556)	2919	39060	8577
Train + SR + RS + RD - (67408)	3892	52080	11436
Train + SR + RS + RI - (67408)	3892	52080	11436
Train + SR + RI + RD - (67408)	3892	52080	11436
Train + RI + RS + RD - (67408)	3892	52080	11436
Train + SR + RS + RD + RI - (84260)	4865	65100	14295

C. Data Augmentation

The data augmentation technique employed in this paper is called Easy Data Augmentation (EDA), it consist of four primary operations which are: Synonym replacement (SR), Random Swap (RS), Random Deletion (RD) and Random Insertion (RI). They are explained briefly as follows:

- **Synonym replacement (SR):** Given a text data set, this operation goes through each sentence in the corpus, selecting a number of words "n" at random and replacing each selected word with one of its own random synonyms as well. In addition stop words like 'a' & 'and' etcetera are not selected as words.
- **Random Swap (RS):** It iterates through each sentence selecting two words randomly and swapping their positions, this is done n times.
- **Random deletion (RD):** It goes through each sentence in the corpus, with a probability "p" each word in the sentence is removed or not based based on p
- **Random Insertion (RI):** Similarly to SR, this operation goes through each sentence in the corpus, selecting a number of words "n" at random and adding a synonym

of each selected word at random positions in the sentence, again stop words like ‘a’ & ‘and’ etcetera are not selected as words.

As each sentence in the dataset corpus varies, both long and short sentences are present in the corpus. Long sentences can absorb more noise and still hold their assigned class label unlike short sentences, this is because short sentences contain fewer words. As the aim of text augmentation is to create new data samples which still possess their class labels after undergoing the various augmentation operations. The numbers of words n changed for the operations SR,RS,RI is set to vary dependent on the length of the sentence, where ($n = \alpha l$), l being the sentence length, α determining how many many words undergo the operations. The same applies to RD where probability ($p = \alpha$)

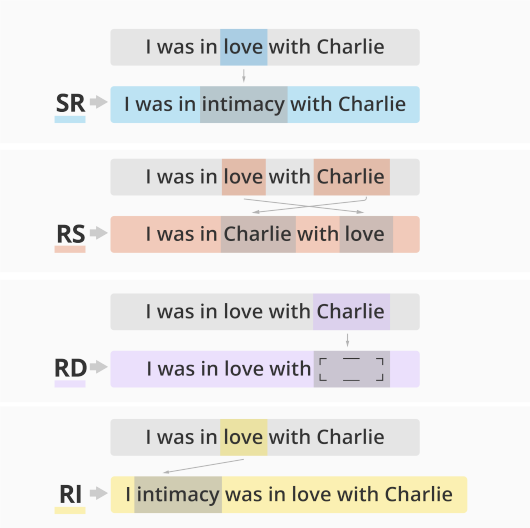

Fig. 1. Augmented sentence on EDA operations

The diagram above graphically illustrates how each EDA technique works. With each operation nuanced to solve a distinct problem. Synonym replacement enables augmented sentences to preserve semantic meaning similar to the source text and equally provides new words to work with. Perturbations are inserted into sentences by the deletion and addition of words from Random swapping and Random Insertion while keeping all words present in the source text, this assist with generalisation. Over-fitting is mitigated by Random Deletion which removes words at random prevents our model from learning the underlying training data patterns.

D. Experimental Setup

Experimental Setup

1) Preprocessing: Data preprocesssing is a necessary step required to clean data and get rid of unnecessary features which would count as noise to model. This for example could include stop words, hashtag signs, numbers etc. Following this the data is split using a train_test_validation split. The split data obtained is saved into seperate csv files to avoid having

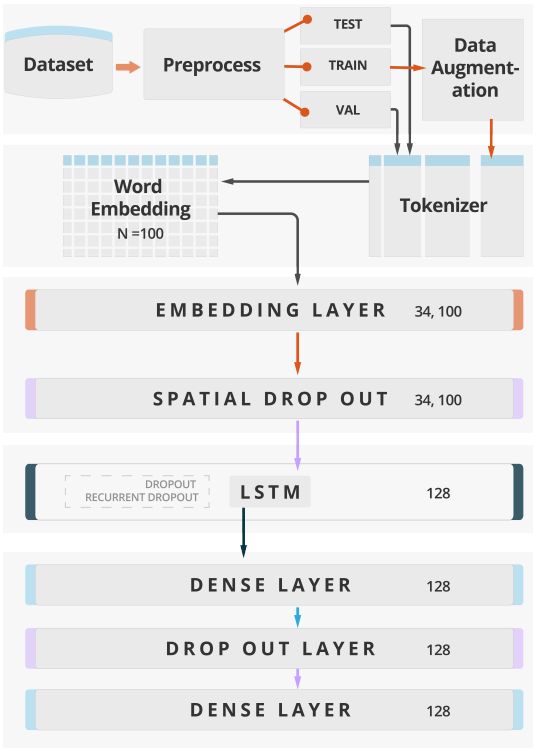

Fig. 2. Hate Speech classification pipeline

new values in the train, test and validation set every time the algorithm program file is executed.

2) Data Augmentation: Here the training data specifically undergoes augmentation with the use of the four augmentation methods and various combinations of them to increase the training data corpus size. As they are four, fifteen unique combinations are obtained.

3) Tokenization: At this stage our y label values for the train, test and validation split are converted to to categorical values. Our X values for the train, test and validation which are the tweets possess sentences, each sentence containing a list of words. The tokeniser makes a vocabulary of unique words which it then uses to represent each word by their word id. The sequences obtained are padded to make sure the input that goes into the LSTM network are of the same length.

4) Embedding: Here a pretrained Glove word embedding with a 100 dimensions is used for transfer learning . Its trainable parameter is set to false, with this the Glove word embeddings are used as initialisation for the embedding layer of the simple LSTM. An index of words(embedding matrix) is created based on the glove database with 100 dimensions and from this a word vector embedding

5) LSTM Architecture: The LSTM model architecture consist of 6 layers, 1 embedding layer, 1 spatial drop out layer, a 128 neuron LSTM layer with input dropout and recurrent drop out, and 1 dense layer with relu activation function, 1 dropout layer and an output dense layer with a soft max function. Using the embedding matrix created, we initialise the weights of the LSTM networks embedding layer, training is set to false so

the embedding is not updated rather the the LSTM network uses the embedding which is the embedding matrix. With this the model train fast using existing vector representation.

IV. EXPERIMENTATION & RESULTS

This section evaluates the results obtained from the LSTM model trained on 16 different datasets, with 1 original source text sample, and 15 distinct augmented data samples. Its subsection would contextualise the base model as a benchmark comparison for the augmentation, briefly describe evaluation metrics employed for assessing performance, highlight various methods explored for hyper-parameter, embedding and neural network optimisation, after which analyse results obtain from the combination of easy data augmentation techniques.

A. Base model as a baseline

As this paper has no directly similar works utilising a hate-speech dataset, closely related comparisons can't be made with other papers. Nonetheless two papers (Davidson et al., 2017) and (Hemker, 2018) which tackled the same multi classification task using the exact same dataset are worth mentioning. Davidson et al. (2017) who has been mentioned consequently in this paper, used statistical methods like linear SVMs, Logistic Regression(LR) and Naive Bayes etcetera settling on LR and achieving an F1 score of 0.91. Hemker (2018) who's work followed after, experimented with numerous deep learning methods and some computationally expensive data augmentation techniques achieving F1 score of 0.94. While these works focused on achieving state of the art results outperforming previous benchmarks, this paper is more concerned with assessing percentage improvement of a model classifier when augmentation is used on an imbalanced dataset and not used. Hemker (2018) explored augmentation as an added appendage to other attempted avenues their best augmentation result using threshold synonym replacement gave a 2 to 3% increase in their previous benchmark F1 score of 0.94 raising it to 0.97

B. Evaluation Metrics

The goal of a given machine learning task as well as challenges faced determine the metrics used to it to assess it. In this case an imbalanced dataset problem, involving multi class classification, where the minority class is essential to assessment is being tackled. As such, accuracy which refers to the total number of correctly made predictions divided by the total number of predictions is not sufficient, this is because in an instance where the majority class is more prominent, the minor class may not be properly represented. The metrics used, present a wholesome view on the performance of the model. These are precision, recall, F1 Score, and confusion matrix, they are described briefly below.

- Precision
  Precision is also known as positive predictive value, quantifies the number of positive classified samples which are correct positive. Where: Precision = (TP/TP +FP)
- Recall
  Recall also known as Sensitivity/True Positive Rate(TPR),quantifies the number of positive Classified samples which were correctly predicted as positive by the model. Where: Recall = (TP/TP + FN)
- F1 Score
  F1 Score also known as F-Measure, refers to the harmonic mean of both precision and recall, with both values combined to form a single metric. Where: F1 Score = (2×Precision × Recall / (Precision + Recall))

C. Optimisation methods explored

Optimisation method refers to experimentation performed on a trial and error basis to select the most suitable building blocks for the multi class problem. It briefly highlights the hyper parameters, neural network architecture and embedding explored.

1) Hyper parameters: These features where tuned via manual trials runs for each parameter. A spatial dropout, one input Dropout and recurrent dropout as well as a separate dropout layer are used. Different values where experimented with to reduce variance and stop overfitting. Various batch sizes form 32 to 2048 where tested, a size of 128 was used as it allowed the optimum capture of sparsely available minority class data in every batch epoch run. For weighting, class weight initialisation worked best as it evenly distributed the weighting based on class percentage representation in the dataset. Learning rates from 0.01 to 0.00001 where tested with (0.00001) performing best. Epochs starting from a value of 25 where also tried out, with 100 epoch achieving the best results. Categorical cross-entropy was chosen as loss function after few comparisons and Adam was used as the optimiser after contrasts with SGD.

2) Embedding: Word embedding refer to dense vector representations for words, three word embedding package where tried out and "Wikipedia 2014 + Gigawords 5" a GloVe embedding performed best. This was further used in a pretrained and retrained fashion with both results being quite similar but pretrained embedding providing double the speed of the latter. Hnece hence pretraining method was chosen

D. Augmentation Results using EDA Techniques

An important question for linguistic based augmentations(synonyms, nouns) is how to go about performing the operations, which works best? As the unique advantages of each were touched on in chapter 3 Here the results obtained from the four EDA techniques as well as their concatenated combinations are evaluated. The augmented datsets are grouped based on the number of augmentation techniques used at once on the source dataset. They are grouped into -

- Single or no augmentation:
  (LSTM, LSTM+SR, LSTM+RS, LSTM+RD, LSTM+RI)
- Double augmentation:
  (LSTM+SR_RS, LSTM+SR_RD, LSTM+SR_RI, LSTM+RS_RD, LSTM+RS_RI, LSTM+RD_RI)

- Three or more augmentations: (LSTM+SR_RS_RD, LSTM+SR_RS_RI, LSTM+SR_RI_RD, LSTM+RI_RS_RD, LSTM+SR_RS_RD_RI)

1) Single or no Augmentation: Table II on the next page shows interesting findings with relation to single augmentation methods. (LSTM+RS) achieves the highest accuracy and precision, (LSTM+RD) the best recall and hate recall, (LSTM+RI) the highest F1 score, while (LSTM+SR) obtains lowest accuracy, precision and recall score of the four augmented methods excluding the base model (LSTM)

As observed from Fig 3, in the initial training phase (LSTM+SR),(LSTM+RD) and the base model (LSTM) began at epoch 0 with an F1 score lower than 35 on the other hand, (LSTM+RS) which achieved the best accuracy and recall started with a score quite close to 40. The base model (LSTM) shows an immediate divergence from the augmented methods gaining momentum quite slowly, at epoch 20 the base model has an F1 score of around 47 with all the other methods having a score above 55. Equally at this point(epoch 20), (LSTM+SR) begins to diverge as well rising with a slower pace as the epochs progress. (LSTM+RS), (LSTM+RD), (LSTM+RI) trend upwards at a similar rate with less than a difference of 3 between them F1 score wise. (LSTM+RI) achieves the best F1 score maintaining the highest value through majority of the epochs.

The table gives the more insight into as how it achieves the highest F1 score with a 12.38% improvement on the base model(LSTM), while (LSTM+RI) doesn't achieve the best score in any other category it's observed it has the second best in precision, recall, and accuracy. High accuracy with high precision signal good proper handling of the classification task, although it achieves an improvement of 20.41% on the base model with hate recall, which is in the mid range when compared to other augmented hate recall improvement gains. (LSTM+RD) achieves both the highest recall and hate recall score making gains of 11.76% and 25% respectively on the base model at epoch 98 while its F1 score was second best in the table. (LSTM+RS) while having the highest accuracy and precision with gains of 9.84% and 13.74% respectively obtained the lowest hate recall of the augmented methods with an increase of 11.36% on the base model.

2) Double Augmentation: Table III below shows insights with regards to double augmentation methods. (LSTM+RD_RI) achieves the highest accuracy and precision, (LSTM+RS_RD) the best recall and hate recall, while (LSTM+RS_RI) the highest F1 score topping (LSTM+RS_RD) on this metric by a value of 0.04

As illustrated in Fig 4, at the initial training phase (LSTM+RD_RI) began at epoch 0 with an F1 score value lower than 20, on the other hand, (LSTM+RS_RD) which achieved the best recall and hate recall started with a score right above 45, all other double augmented methods had F1 Score values within the range of 40 to 44. Roughly, at the 25th epoch (LSTM+SR_RD) and (LSTM+SR_RI) begins to diverge slightly progressing upwards at a slower pace compares to the rest. (LSTM+RS_RD) in contrast at around the 55th epoch begins to trend upwards faster than its counterparts, although (LSTM+RS_RI) catches up at around the 90th epoch and overtakes at the 98th epoch where it achieves its optimum value, out performing (LSTM+RS_RD) marginally.

Similarly, table III values for (LSTM+RS_RI) allude to why it has the best F1 Score with a 15.76% increase on the base model, its precision and recall fall within the top three in table III. And like its peer in the preceding table I (LSTM+RI) it would perform quite well with classification task. (LSTM+RS_RD) achieves both the highest recall and hate recall score making gains of 16.1% and 32.76% respectively on the base model at epoch 96 while its F1 score was also the second best in table III. Equally (LSTM+RD_RI) while possessing the best accuracy and precision with gains of 11.95% and 16.53% respectively got the lowest hate recall of the double augmented methods with an improvement of 11.36% on the (LSTM).

3) Three or more Augmentation: Table IV exhibits insights discovered with relation to the three or more augmentation category. (LSTM+RI_RS_RD) triple augmentation method achieves the highest accuracy, precision, recall and F1 Score but coupled with that the lowest hate recall of the table. In contrast (LSTM+SR_RI_RD) obtained the best hate recall in the table. As illustrated in Fig 5, at the initial training phase (LSTM+SR_RS_RD_RI) began at epoch 0 with an F1 score value right above 45 , on the other hand, (LSTM+SR_RS_RI) started with F1 score around 20, all other double augmented methods had F1 Score values within the range of 40 to 45. (LSTM+RI_RS_RD) begins to diverge upwards at around the 8th epoch, (LSTM_SR_RS_RD_RI) maintained the highest F1 Score till about epoch 20 where (LSTM+RI_RS_RD) catches up. They both pull away from their peers by a difference of 1 F1 Score wise, trending upwards faster.

(LSTM+SR_RS_RI) has best accuracy, precision, recall, and F1 Score with improvement values of 12.26%, 18.1%, 16.8% and 17.5% respectively on the base model. (LSTM+SR_RI_RD) achieves a 29.1% increases on the base model. In addition its worth noting that (LSTM+SR_RS_RD_RI) the quadruple augmentation method obtained the second best scores in accuracy, precision and F1 Score.

Overall its observed that there is 9-20% improvement across the highest metrics of single augmentation on the base model, a 3-9% improvement comparing the highest metrics of single augmentation to double augmentation and a 1-3% increase comparing the double augmentation methods to the triple and quadruple augmentation methods. There is a similar correlation between high recall and high hate recall across table I to II equally there's also a relation between high accuracy and low hate recall. (LSTM+SR_RS_RI) a triple augmentation, obtained the highest accuracy, precision,

TABLE II
SINGLE DATA AUGMENTATION TECHNIQUES

Model	Best Accuracy epoch	Accuracy	Precision	Recall	F1	Hate Recall
LSTM	99	74.10	53.88	62.27	57.77	0.39
LSTM + SR	99	78.45	59.46	66.7	62.87	0.51
LSTM + RS	97	**82.19**	**62.46**	67.80	65.02	0.44
LSTM + RD	98	78.87	60.70	**70.57**	65.27	**0.52**
LSTM + RI	100	81.14	62.09	70.28	**65.93**	0.49

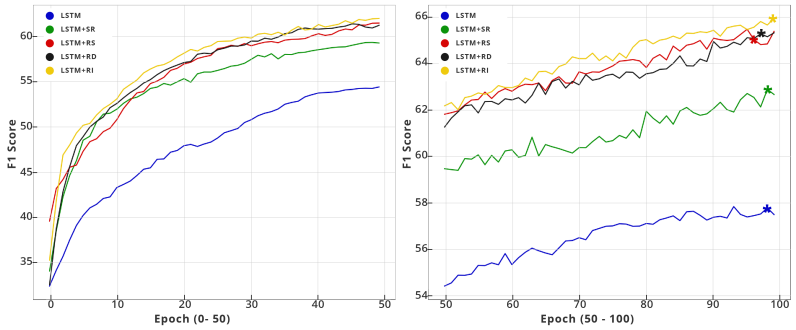

Fig. 3. F1 Score(Single augmentation)

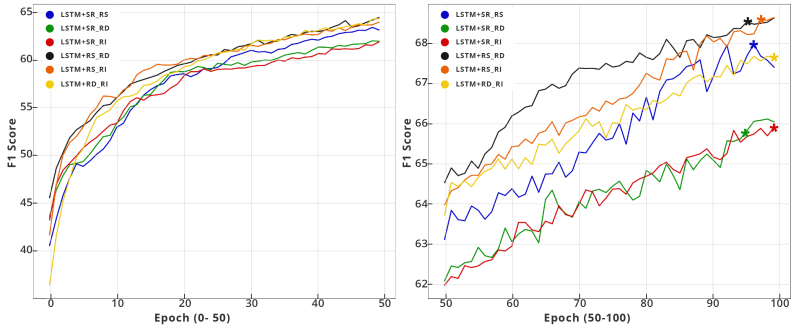

Fig. 4. F1 Score(2 augmentations)

TABLE III
2 JOINT DATA AUGMENTATION TECHNIQUES

Model	Best Accuracy epoch	Accuracy	Precision	Recall	F1	Hate Recall
LSTM + SR_RS	97	81.66	63.77	72.78	67.98	0.57
LSTM + SR_RD	96	81.67	62.46	69.58	65.83	0.47
LSTM + SR_RI	100	82.45	62.94	69.17	65.91	0.45
LSTM + RS_RD	96	81.44	63.67	**74.22**	68.54	**0.58**
LSTM + RS_RI	97	82.81	63.92	73.97	**68.58**	0.52
LSTM + RD_RI	100	**84.16**	**64.55**	71.05	67.64	0.44

recall and F1 Score(12.26%, 18.1%, 16.8% and 17.5% improvement) all at once. Although the best hate recall was recorded with (LSTM+RS_RD) getting an improvement of 32.76% on the base model.

V. DICUSSIONS/CONCLUSIONS

This section has a conversation about the results and ideas mentioned in the previous chapters highlighting insights and problems faced along the way. Its subsections summarises crucial findings discovered from experimentation, discuss challenges faced when during the project duration, highlight limitations faced with the techniques and source materials employed and concludes

A. Findings

The project entailed the implementation of four EDA operations for data augmentation each with a distinct ability, SR replaced synonyms, RS swapped words, RI inserted words which were synonym of existing words in the sentence and RD deleted random words based on a probability. These

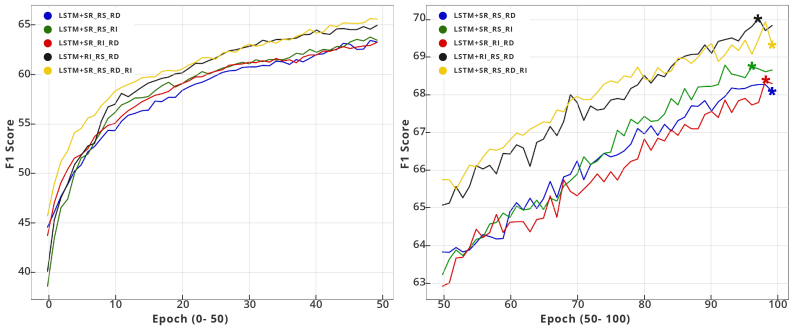

Fig. 5. F1 Score(3 augmentations)

TABLE IV
3 OR MORE DATA AUGMENTATION TECHNIQUES

Model	Best Accuracy epoch	Accuracy	Precision	Recall	F1	Hate Recall
LSTM + SR_RS_RD	100	81.90	63.60	73.26	68.09	0.54
LSTM + SR_RS_RI	98	82.68	64.34	73.66	68.70	0.54
LSTM + SR_RI_RD	99	82.11	64.33	72.94	68.37	**0.55**
LSTM + RI_RS_RD	98	**84.45**	**65.79**	**74.84**	**70.02**	0.52
LSTM + SR_RS_RD_RI	100	83.70	65.69	73.39	69.34	0.54

various operations where concatenated together in different combinations. Each augmented dataset was used to train the model coupled with pretrained word embeddings. The insights discovered are as follows:

1) Data Augmentation improves model performance, and combats imbalance: Results from the previous section show an increase of more than 30% (LSTM+RS_RD) over the base model in hate speech recall , hate(0) being an underrepresented class, of strong importance to the classification task. Equally the precision, recall, and F1 Score collectively rose by more 16% this was achieved the (LSTM+RI_RS_RD) augmentation method. These results exhibit how effective data augmentation is to an imbalanced multi class NLP problem.

2) High model Accuracy doesn't guarantee good model performance: Based on several experiments a trend was observed from the single, double and three or more augmentation tables. The augmentation methods which achieved the highest accuracies subsequently had the lowest hate speech recalls in their different tables respectively. Hence as a result shows accuracy with imbalanced datasets are not solely sufficient.

B. Contributions

As mentioned in an earlier chapter 2, data imbalance is readily present in real world scenarios. In some cases it properly mirrors the very nature of the data(the fact it appears sparsely). Attempting always get sufficient data samples can be cumbersome and less cost effective. Data augmentation is an appropriate solution to mitigate this.

C. Challenges

These details encapsulate the roadblocks encountered and problem discovered while working on the project. They are as follows:

1) (OOV) words for Word net: An important challenge for EDA techniques was how they worked with out of vocabulary(OOV) words. SR and RI both use synonyms to perform their operations, when words which were derogatory(hate speech) and spelt wrongly to mask intent were encountered for example (n*igga vs n*cca) in such instances synonyms could be found from the lexical database to replace such words. This overall could have reduced performance.

2) (OOV) words for Embedding: As pretrained embeddings where used for the classification task, the dense vector representation which GloVe presented couldn't adequately represent OOV words as well. This is especially prevalent as mentioned earlier with words spelt wrongly in some cases to mask meaning and pass on a hateful or offensive message subtly.

D. Limitations

Although EDA has made a strong contribution to data augmentation for NLP there are still limitations it encounters that should be addressed which are discussed as follows

1) Language barrier for Lexical Database: EDA as a text augmentation technique performs four different operations two of its operations Synonym replacement and Random Insertion strongly rely on the use of a lexical data base called WordNet which is in English. It implies that currently, EDA is only a locally applicable solution and can't be used for tackling natural language processing problems with different languages which lack lexical databases. If EDA must be applied, such a database would have to be built this would come at a relatively high cost

2) Hate speech Dataset: The dataset itself was one of the biggest limitations to the project itself. A few spotted

instances of labelled data were wrongly labelled by the human encoders. This implies that the data set is equally subject to human bias which could affect performance. As such it would need to be factored for experimentation, an appropriate solution would be to utilise and make comparisons between different datasets for the same task but with at hand, to my knowledge only one or two publicly available multi class hate speech related datasets exist

In conclusion, we began with a premise of testing an idea, "Can data augmentation improve performance with an imbalanced dataset ?", following this, to test this theory we employed the use of a universal group of augmentation techniques called EDA coupled with pretrained embedding and an RNN classifier. This techniques vastly improved performance and acted as means of regularisation reduce the model probability of over fitting. With these techniques weak points were spotted such as the language barrier limitation and the handling of OOV words. These underlying problems give room for future work.

VI. Future Work

The paper experimented with a fascinating idea to solve data imbalance in a textual dataset while improving classification performance, all this using combination mix of 4 augmentation methods which relied on a lexical database. As EDA primarily works with synonyms swapping, inserting, randomly deleting and replacing words, Interesting idea would be as follows:

- To develop an EDA technique which could swap synonyms of words with antonyms and then change their labels. This would further assist creating even more varied dataset

- As EDA primarily work with linguistic features such as synonyms a more context specific lexical data bases can be developed working with social media platforms who are at the funnel for mass public text based communication online.

References

T. Alan. "anti-muslim hate crime surges after manchester and london bridge attacks", Jun 2017. URL https://www.theguardian.com/society/.

O. Alexis. "hate on the rise after trump's election", Nov 2016. URL www.newyorker.com.

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

P. Burnap and M. L. Williams. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, 7 (2):223–242, 2015.

T. Davidson, D. Warmsley, M. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*, 2017.

M. Fadaee, A. Bisazza, and C. Monz. Data augmentation for low-resource neural machine translation. *arXiv preprint arXiv:1705.00440*, 2017.

T. B. I. for Global Community. "designating hate: New policy responses to stop hate crime", Aug 2019. URL https://institute.global/policy.

K. Hemker. *Data augmentation and deep learning for hate speech detection*. PhD thesis, Master's thesis, Imperial College London, 2018.

N. Jaitly and G. E. Hinton. Vocal tract length perturbation (vtlp) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, volume 117, 2013.

T. Ko, V. Peddinti, D. Povey, and S. Khudanpur. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

S. Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*, 2018.

O. Kolomiyets, S. Bethard, and M.-F. Moens. Model-portability experiments for textual temporal analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, volume 2, pages 271–276. ACL; East Stroudsburg, PA, 2011.

B. Krawczyk, M. Galar, Ł. Jeleń, and F. Herrera. Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. *Applied Soft Computing*, 38:714–726, 2016.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

I. Kwok and Y. Wang. Locate the hate: Detecting tweets against blacks. In *Proceedings of the twenty-seventh AAAI conference on artificial intelligence*, pages 1621–1622, 2013.

N. Lomas. "facebook, twitter still failing on hate speech in germany as new law proposed", Mar 2017. URL www.techcrunch.com.

J. Mueller and A. Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *thirtieth AAAI conference on artificial intelligence*, 2016.

L. Natasha. "facebook, google, twitter commit to hate speech action in germany", Dec 2015. URL www.techcrunch.com/2015.

M. Nicky. "update on online harms:written statement - hlws107", Feb 2020. URL www.parliament.uk/business/publications/.

R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.

S. Sharma, S. Agrawal, and M. Shrivastava. Degree based classification of harmful speech using twitter data. *arXiv preprint arXiv:1806.04197*, 2018.

S. Shleifer. Low resource text classification with ulmfit and backtranslation. *arXiv preprint arXiv:1903.09244*, 2019.

T. Simonite. "facebook's ai for hate speech improves. how much is unclear", May 2020. URL `https://www.wired.com/`.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

S. Theil. The online harms white paper: comparing the uk and german approaches to regulation. *Journal of Media Law*, 11(1):41–51, 2019. doi: 10.1080/17577632.2019.1666476. URL `https://doi.org`.

D. van Mill. Freedom of Speech. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2018 edition, 2018.

V. Van Vlasselaer, C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, and B. Baesens. Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*, 75:38–48, 2015.

W. Y. Wang and D. Yang. 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, 2015.

X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.